日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

#3
3-12-01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年 ６月 ８日

出 願 番 号
Application Number: 特願２０００－１７７１２３

出 願 人
Applicant (s): 株式会社日立製作所

U.S. Appln Filed 12-20-00
Inventor. K. Serizawa et al
Mattingly Stanger Malur
Docket 111-245

CERTIFIED COPY OF PRIORITY DOCUMENT

２０００年１１月１０日

特 許 庁 長 官
Commissioner,
Patent Office 及 川 耕 造

出証番号 出証特２０００－３０９２８６３

【書類名】　特許願

【整理番号】　NT00P0190

【提出日】　平成12年 6月 8日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 13/38

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　芹沢　一

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　長須賀　弘文

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　櫻庭　健年

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　二瀬　健太

【発明者】

【住所又は居所】　神奈川県秦野市堀山下１番地　株式会社日立製作所　エンタープライズサーバ事業部内

【氏名】　山下　正弘

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社日立製作所

【代理人】

【識別番号】　100068504

【弁理士】

【氏名又は名称】　小川　勝男

【電話番号】　03-3661-0071

【選任した代理人】

【識別番号】　100086656

【弁理士】

【氏名又は名称】　田中　恭助

【電話番号】　03-3661-0071

【選任した代理人】

【識別番号】　100094352

【弁理士】

【氏名又は名称】　佐々木　孝

【電話番号】　03-3661-0071

【手数料の表示】

【予納台帳番号】　081423

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】計算機システムおよびそのデータ転送方法

【特許請求の範囲】

【請求項１】第１の計算機ノードとそれに接続された第２の計算機ノードを有する計算機システムであって、第１の計算機システムにあって、データレコードを格納する第１の記憶領域と、任意の時間間隔で第２の計算機ノードとは非同期に第１の記憶領域にデータレコードを格納する第１の処理部と、第２の計算機ノードにあって第１の記憶領域から転送されたデータレコードを記憶する第２の記憶領域と、第１の計算機ノードとは非同期の任意の時間間隔で第１の記憶領域から読み出すべきレコード群を指定して第２の記憶領域に読み出し、参照する第２の処理部を備えたことを特徴とする計算機システム。

【請求項２】第２の計算機ノードは第２の処理部を一定の時間間隔で起動し、第１の記憶領域から第２の記憶領域へのデータの読み出しを行なわしめるタイマを備えたことを特徴とする請求項１記載の計算機システム。

【請求項３】第１の処理部はデータレコードが格納された順序を示す識別番号を付与して前記データレコードを第１の記憶領域に格納し、第１の記憶領域は前記識別番号とデータレコードが組で格納される複数個のエントリを有しており、前記エントリに第１の処理部で書き込まれる方向と前記エントリから読み出される方向は逆であり、第２の処理部は第２の記憶領域に転送された第１の記憶領域のデータを参照し、前記識別番号によって当該データレコードが正しいものであるか否かを判断することを特徴とする請求項１記載の計算機システム。

【請求項４】第１の処理部では前記データレコードを書き込んでから当該データレコードの識別番号を書き込み、第２の処理部では第２の記憶領域に読み出されたデータの前記識別番号に連続性がある場合当該データレコードは正しいと判断し、連続性がない場合当該データレコードが不正であると判断することを特徴とする請求項３記載の計算機システム。

【請求項５】第１の処理部は更に前記データレコードに対して誤り検出符号を生成する誤り検出符号生成処理部を含み、第１の記憶領域に前記データレコードと前記誤り検出符号を書き込み、第２の処理部は第２の記憶領域に読み出された

データについて前記誤り検出符号により誤りの検出を行ない、誤りが検出されない場合当該データレコードを正しいものと判断し誤りが検出された場合当該データレコードを正しくないものと判断することを特徴とする請求項１記載の計算機システム。

【請求項６】第１の記憶領域は前記誤り検出符号とデータレコードが組で格納される複数個のエントリを有しており、前記エントリに第１の処理部で書き込まれる方向と前記エントリから読み出される方向とは同一方向であることを特徴とする請求項５記載の計算機システム。

【請求項７】第１の計算機ノードと第２の計算機ノードが接続された計算機システムであり、第１の計算機ノードはデータレコードを格納する第１の記憶領域と、任意の時間間隔で第２の計算機ノードと非同期に第１の記憶領域に前記データレコードを格納する第１の処理部と、任意の時間間隔で第１の記憶領域の前記データレコードを第２の計算機ノードに送信するデータ送信要求を生成するデータ送信生成処理部とを有し、第２の計算機ノードは転送された第１の記憶領域のデータレコードを格納する第２の記憶領域と、第２の記憶領域のデータレコードを任意の時間間隔で第１の計算機ノードと非同期に参照する第２の処理部を備えたことを特徴とする計算機システム。

【請求項８】第１の計算機ノード上の第１の記憶領域に格納された一つ以上のレコードで構成されるデータを、第２の計算機ノード上で動作しているプログラムが、当該記憶領域を指定して、第２の計算機ノード上の第２の記憶領域に転送せしめて参照することが可能な通信手段を有する計算機システムのデータ転送方法において、第１の計算機ノード上で動作し、任意の時間間隔で、第１の記憶領域に、一つ以上のレコードで構成されるデータを格納するステップと、第２の計算機ノード上で動作し、任意の時間間隔で、前記通信手段を用いて第１の記憶領域の指定されたデータを第２の記憶領域に転送せしめ、そのデータを参照するステップとを備えたことを特徴とするデータ転送方法。

【請求項９】第１の記憶領域を持つ第１の計算機ノード上で動作しているプログラムが、第２の計算機ノードの主記憶装置内の第２の記憶領域に、第１の記憶領域の一つ以上のレコードで構成されるデータを、直接格納することが可能な通

信手段を有する計算機システムのデータ転送方法において、第１の計算機ノード上で動作し、任意の時間間隔で、前記通信手段を用いて第２の記憶領域に一つ以上のレコードで構成されるデータを格納するステップと、第２の計算機ノード上で動作し、任意の時間間隔で、第２の領域のデータを参照するステップとを備えたことを特徴とするデータ転送方法。

【請求項１０】第１の記憶領域は識別番号とデータレコードが組で格納される複数個のエントリを有しており、第１の計算機ノード上で動作し、前記データレコードを書き込んでから当該データレコードの前記識別番号を書き込み、前記エントリに書き込まれる方向と逆の方向に前記エントリから読み出すステップと、第２の計算機ノード上で動作し、第２の記憶領域に転送された第１の記憶領域のデータを参照し、第２の記憶領域に読み出されたデータの前記識別番号に連続性がある場合当該データレコードは正しいと判断し、連続性がない場合当該データレコードが不正であると判断するステップとを備えたことを特徴とする請求項８または９記載のデータ転送方法。

【請求項１１】第１の記憶領域は誤り検出符号とデータレコードが組で格納される複数個のエントリを有しており、第１の計算機ノード上で動作し、第１の記憶領域に前記データレコードとその誤り検出符号を書き込み、前記エントリに書き込まれる方向と同一方向に前記エントリから読み出すステップと、第２の計算機ノード上で動作し、第２の記憶領域に読み出されたデータについて前記誤り検出符号により誤りの検出を行ない、誤りが検出されない場合当該データレコードを正しいものと判断し誤りが検出された場合当該データレコードを正しくないものと判断するステップとを備えたことを特徴とする請求項８または９記載のデータ転送方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、ネットワークまたは入出力チャネルで接続された計算機の間で、複数のレコードからなるデータを受け渡す処理に関する。

【０００２】

【従来の技術】

二つの計算機ノード間で、多数のレコードからなるデータを受け渡す従来の技術として、以下に示すものがある。

【０００３】

第１の従来の技術は、特開平６－６７９４４号公報の２ページ６１行目から、２ページ８１行目にに示されている。この第１の従来の技術は、２つの計算機ノード間で共用しているディスク装置を利用した方法である。この方法では、同一のデータを格納した２個のボリュームにからなる組を用意し、各々のボリュームを各計算機ノードに接続させて、共用できる状態にしておく。そして、一方の計算機ノードがデータを参照する場合は、ボリュームの組を解き（ボリュームの切り離し）、一方のボリューム（以下、第１のボリューム）を参照する側の計算機ノードに占有させる。その間、該ディスクの制御装置は、もう片方のボリューム（以下、第２のボリューム）に他方の計算機ノードによる変更を全て記録しておく。データを参照した計算機ノードが、参照を終え、上記第１のボリュームの占有を解くと、上記ディスクの制御装置は、上記第２のボリュームに対する変更の記録を、上記第１のボリュームに反映し、その後、上記２個のボリュームを、同一のデータを格納する組として、二つの計算機ノードから共用できる状態にする（ボリュームの再同期化）。

【０００４】

第２の従来の技術は、特開平６－１４９４８５号公報の３ページ５８行目から、４ページ５２行目に示されている。この第２の従来の技術は、計算機ノード間で共用している半導体外部記憶装置を利用した方法である。この方法では、単一のメモリ領域を複数の計算機ノード間で共用し、各々の計算機ノードは、該メモリ領域に対して排他的にアクセスする。

【０００５】

【発明が解決しようとする課題】

ここで、第１の従来の技術は、一方の計算機ノードがデータを参照するたびにボリュームの切り離しと、再同期化を行う必要が有る。そのため、リアルタイム

処理に適用することは困難である問題がある。

【０００６】

一方、第２の従来の技術は、データ転送のたびに、レコードの完全性を保証するためにデータの出力側の計算機ノードと、データの参照側計算機ノードとの間で、これらの領域の排他制御を行う必要がある。大量のデータを転送する際には、これらの排他処理に要するＣＰＵオーバヘッドが膨大となる問題がある。さらに、このＣＰＵオーバヘッドは、データ転送の効率を低下させることがある。

【０００７】

本発明の第１の目的は、データ転送効率の向上を阻害する排他処理に要するＣＰＵオーバヘッドを軽減することにある。

【０００８】

本発明の第２の目的は、リアルタイム処理にも利用可能なデータ転送方法を提供することにある。

【０００９】

【課題を解決するための手段】

本発明では、ＲＤＭＡ（Ｒｅｍｏｔｅ　Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ）を用いた、データ転送を行う。ＲＤＭＡとは、送信側の計算機ノードの送信すべきデータのアドレスを受信側の計算機ノードが判っている、または受信側の計算機ノードの受信すべきデータのアドレスを送信側の計算機ノードがわかっているものである。そして、ネットワークで接続された二つの計算機ノード間において、一方の計算機ノードのプログラムが、該計算機ノードの主記憶のデータを格納すべき／データを読み出すべき領域と、他方の計算機ノードの主記憶のデータを読み出すべき／データを格納すべき領域とを指定してそれらの領域間でデータのコピーをする要求を作成し、この要求を、通信手段またはそれを制御するソフトウェアで処理することで、上記計算機ノードのそれぞれの主記憶間で直接データのコピーを行う技術である。

【００１０】

ＲＤＭＡには、ＲＤＭＡを起動する計算機ノードの主記憶のデータを、他方の計算機ノード上の主記憶に格納するＲＤＭＡ－Ｗｒｉｔｅと、ＲＤＭＡを起動す

る計算機ノードの主記憶に、他方の計算機ノード上の主記憶のデータを格納するＲＤＭＡ－Ｒｅａｄの２種がある。

【００１１】

ＲＤＭＡは、例えば米Ｉｎｔｅｌ社、米Ｃｏｍｐａｑ社、米マイクロソフト社による、Ｖｉｒｔｕａｌ　Ｉｎｔｅｒｆａｃｅ　Ａｒｃｈｉｔｅｃｔｕｒｅ　Ｓｐｅｃｉｆｉｃａｔｉｏｎ　１．０（１９９７．１２．１６）に記載されている。

【００１２】

本発明では、第１の計算機ノードの主記憶上の領域に、同期を取らずに一方的に（非同期の手順で）任意の時間間隔で、一つ以上のレコードを格納し、第２の計算機ノード上で動作しているプログラムが、ＲＤＭＡ－Ｒｅａｄを利用して、当該領域を任意の時間間隔で参照することで、データ転送を実現する。

【００１３】

また、第１の計算機ノード上のプログラムが、第２の計算機ノードの主記憶上の領域に、同期を取らずに一方的に（非同期の手順で）任意の時間間隔で、ＲＤＭＡ－Ｗｒｉｔｅを用いて一つ以上のレコードを格納し、第２の計算機ノード上で動作しているプログラムが、上記領域を任意の時間間隔で参照することでデータ転送を実現する。

【００１４】

【発明の実施の形態】

以下、本発明の実施の形態を、図を用いて説明する。

まず、図１から図１０を用いて第１の本発明の実施の形態を説明する。

図１は、第１の本発明の実施の形態の全体構成図である。第１の計算機ノード１０と第２の計算機ノード２０は、ネットワーク３０に接続している。そして、第１の計算機１０と第２の計算機２０とは、ネットワーク３０を介して互いに通信することが可能である。

【００１５】

第１の計算機ノード１０には、第２の計算機ノード２０に送信すべきデータレコード１５３を出力する第１のプログラム１１０と主記憶１５０に格納され、デ

ータレコード１５３を格納するデータレコードテーブル１５１と、前記データレコード１５３をネットワーク３０を経由して第２の計算機２０へ送信する送信部１７０を有している。これは、第１のプログラムと独立したプログラム、又はハードウェアで構成されている。さらに、第１のプログラム１１０は、前記データレコード１５３を出力するデータレコード出力処理部１１１と、後述する識別情報１５２を出力する識別情報出力処理部１１２と、データレコード出力処理部１１１および識別情報出力処理部１１２の出力をデータレコードテーブル１５１に格納する格納処理部１１３で構成される。ここで、前記識別情報１５２とは、少なくとも前後に連続して格納された２個のデータレコード１５３を識別できる情報であり、例えばデータレコード１５３のそれぞれに対し採番される通し番号である。識別情報出力処理部１１２には該通し番号を生成するカウンタ１１６を有する。さらに、格納処理部１１３には、データレコードテーブル１５１における格納すべきエントリのインデクス（ある識別情報及びデータレコードをどのエントリに格納すべきかを示すもの）を格納するポインタ１１５を含む。

【００１６】

また、データレコード出力処理部１１１とは例えば、ＯＬＴＰ（Ｏｎ　Ｌｉｎｅ　Ｔｒａｎｓａｃｔｉｏｎ　Ｐｒｏｃｅｓｓｉｎｇ）がジャーナルデータをデータレコードテーブル１５１に書き込むものであり、この例ではデータレコード１５３はオンライン処理におけるジャーナルデータである。

【００１７】

第２の計算機ノード２０は、第１の計算機１０が出力したデータレコード１５３を受信し、参照する第２のプログラム２１０と第１の計算機１０のデータレコードテーブル１５１の、完全または不完全な複製である主記憶２５０上のデータレコードテーブル２５１と、前記データレコード１５３をネットワーク３０を経由して第１の計算機から受信する受信部２７０を有している。さらに、第２のプログラム２１０は、タイマ２１１と、前記データレコード１５３の受信要求を生成するデータ受信要求生成処理部２１２と、データレコード参照処理部２２１を有する。なお、タイマ２１１は一定の時間間隔でデータ受信要求生成処理部２１２を起動するための処理であり、第２のプログラム２１０の外部に存在しても良

い。

【００１８】

さらに、データレコード参照処理部２２１はデータレコードテーブル２５１において、どのエントリを参照すべきかを示すインデクスを格納するポインタ２２５と、読み出された前記識別情報２５２の妥当性を検証する為に使用するカウンタ２２６とを含む。

【００１９】

なお、データレコード１５３を生成するプログラム、およびデータレコード２５３を参照してさらに別の処理を行うプログラムは、本発明とは直接関係ないので、本実施例では省略する。

【００２０】

図２は第１の実施の形態において、第１のプログラム１１０の処理を示すフローチャートである。

まず、データレコード出力処理部１１１が１個のデータレコード１５３を出力し、格納処理部１１３に該データレコード１５３のデータレコードテーブル１５１への格納処理を依頼する（ステップ１１ａ）。格納処理部１１３は、識別情報出力処理部１１２を起動する（ステップ１１ｂ）。識別情報出力処理部１１２は識別情報１５２を出力し、これを格納処理部１１３に返す（ステップ１１ｃ）。格納処理部１１３は、ステップ１１ａのデータレコード１５３とステップ１１ｃの識別情報１５２とを組にして、ポインタ１１５が指すエントリに格納する（ステップ１１ｇ）。その後、格納処理部１１３は、ポインタ１１５をインクリメントし、最大値を超えた場合はラップ処理を行う（ステップ１１ｈ）。識別情報出力処理部１１２および格納処理部１１３の詳細については後述する。第１のプログラム１１０が複数のデータレコード１５３を出力する場合、上記ステップ１１ａからステップ１１ｈを繰り返す。識別情報及びデータレコードは第２の計算機ノードとは関係なくデータレコードが発生する毎にデータレコードテーブル１５１へと格納される。

【００２１】

図３は、第１の実施の形態において、第２のプログラム２１０の処理を示すフ

ローチャートである。

まず、第２のプログラム２１０はデータレコードテーブル２５１を初期化する（ステップ２１ａ）。初期化後のデータレコードテーブル２５１は、後に図７で説明するため、ここでは説明を省略する。次にデータ受信要求生成処理部２１２が参照先としてデータレコードテーブル１５１を、受信先としてデータレコードテーブル２５１を、それぞれ指定したデータ受信要求を生成し、受信部２７０を起動する（ステップ２１ｃ）。即ち、このときにＲＤＭＡ－Ｒｅａｄの起動が行なわれる。ステップ２１ｃにおいて、送信先および受信先には、データレコードテーブル１５１および２５１の全てのエントリ、または一部のエントリ群、のいずれを指定しても良い。望ましくは、前回の最後のデータ転送において、送信部１７０が最後のエントリを読み出した時刻から、今回のデータ転送において、送信部が最初のエントリを読み出す時刻までの間に、格納処理部１１３が格納するエントリ群を指定する。第１の計算機ノード１０の負荷により、上記エントリ群に含まれるエントリ数が変化する場合は、負荷に追随して読み出すエントリ数を増減する。例えば、前回読み出しに失敗したエントリ数が多い場合は、次に読み出すエントリ数を減少させる。

【００２２】

さらに、第２のプログラム２１０は、受信部２７０からの、ステップ２１ｃで発行したデータ転送の完了を待つ（ステップ２１ｄ、ステップ２１ｅ）。さらに、データレコード受信処理部２２１が、データレコードテーブル２５１を参照する（ステップ２１ｆ）。データレコード参照処理部２２１については、後に図１０を用いて説明する。さらに、第２のプログラム２１０は、タイマ２１１に対し、一定時間後にステップ２１ｃから処理を続行することを要求する（ステップ２１ｇ）。ステップ２１ｇにおいて、タイマに要求する時間間隔は任意で良い。リアルタイム性を向上する為に望ましくは、該時間間隔として、今回のデータ転送において、送信部１７０が最後のエントリを読み出した時刻から、次回のデータ転送において、送信部１７０が最初のエントリを読み出す時刻までの間に、格納処理部１１３が一つ以上のエントリを格納可能な時間間隔を指定する。特に、ステップ２１ｆにおいて、今回データ転送した全てのエントリが読み出し可能であ

った場合には、次のデータが既に格納されている可能性が有るので、時間間隔として０を設定することが望ましい。

【００２３】

データ転送効率を向上する為に望ましくは、該時間間隔として今回のデータ転送において、送信部１７０が最後のエントリを読み出した時刻から、次回のデータ転送において、送信部が最初のエントリを読み出す時刻までの間に、格納処理部１１３がデータレコードテーブル１５１の半分に相当するエントリを格納可能な時間間隔を指定する。次に、タイマ２１１が一定の時間後にデータ受信要求生成処理部２１２を起動する（ステップ２１ｂ）。このようにＲＤＭＡに関して第１の計算機ノードにおけるデータの格納と第２の計算機ノードにおけるＲＤＭＡ－Ｒｅａｄによるデータの読み出しがそれぞれ任意の時間間隔で非同期に行なわれるので、これらの間での確認手順が不要であり、プログラムにかかる負担は小さいものとなる。

【００２４】

以下では、ｌ、ｍ、ｎは１を超える自然数とし、ｌとｎ、およびｍとｎはそれぞれ互いに素であるとする。ｎ－１はカウンタ１１６の上限値の意味を持ち、ｍはデータレコードテーブル１５１のエントリ数を、ｌはデータレコードテーブル２５１のエントリ数をそれぞれ示す。

【００２５】

図４は、第１の実施の形態において、識別情報出力処理部１１２の処理を示すフローチャートである。

まず識別情報出力処理部１１２は、カウンタ１１６を０にクリアし（ステップ１１２ａ）、格納処理部１１３からの要求を待つ（ステップ１１２ｂ、ステップ１１２ｃ）。ここで、格納処理部１１３からの要求があると、識別情報出力処理部１１２はカウンタ１１６の値を格納処理部１１３に返す（ステップ１１２ｄ）。ここで、カウンタ１１６の値がｎ－１より小さいか判断し（ステップ１１２ｅ）、該判断が真の場合、カウンタ１１６をインクリメントし（ステップ１１２ｆ）、ステップ１１２ｂからの処理を繰り返す。該判断が偽の場合は、ステップ１１２ａからの処理を繰り返す。

【００２６】

図５は、第１の実施の形態において、格納処理部１１３を示すフローチャートである。

まず格納処理部１１３は、データレコードテーブル１５１を初期化する（ステップ１１３ａ）。初期化後のデータレコードテーブル１５１は後に図８に説明する。さらに格納処理部１１３は、ポインタ１１５を０にクリアし（ステップ１１３ｂ）、データレコード出力処理部１１１からの要求を待つ（ステップ１１３ｃ、ステップ１１３ｄ）。ここで、データレコード出力処理部１１１からの要求があると、格納処理部１１３は、データレコード出力処理部１１１が出力したデータレコード１５３を受け取り（ステップ１１３ｅ）、さらに図２で示したステップ１１ｂを行い、識別情報出力処理部１１２から識別情報１５２を得る（ステップ１１３ｆ）。さらに、格納処理部１１３は、データレコードテーブル１５１においてポインタ１１５が指すエントリに、ステップ１１３ｅで得たデータレコード１５３を格納し（ステップ１１３ｇ）、ステップ１１３ｆで得た識別情報１５２を、該エントリに格納する（ステップ１１３ｈ）。さらにポインタのインクリメントのための処理を行う（ステップ１１３ｋ、ステップ１１３ｌ）。

【００２７】

図６は、第１の実施の形態において、格納処理部１１３のステップ１１３ａの直後の、即ち初期化後のデータレコードテーブル１５１を示している。

【００２８】

データレコードテーブルはエントリ０からエントリｍ－１で成り立っており、それぞれポインタが０からｍ－１をとるときに指すエントリに対応している。格納処理部１１３は各エントリの識別情報１５２を、次のように格納する。格納処理部１１３はエントリ０の識別情報１５２．０に－１を、エントリ１の識別情報１５２．１からエントリｍ－２の識別情報１５２．ｍ－２までは、０、１、２、．．．の順に、０から１ずつ増加させた数を格納する。ここで、もし、識別情報１５２に格納すべき値がｎ－１を超えた場合には、該エントリの識別情報１５２には０を格納し、以降は同様に１ずつ増加させた数を格納する。さらに格納処理部１１３は、エントリｍ－１の識別情報１５２．ｍ－１にはｎ－１を格納する。

各エントリのデータレコード１５３は、適当な初期値で初期化されている。但し、以降に述べるデータレコード参照処理部２２１では、これらのデータレコード１５３を無視するため、必ずしも初期化する必要はない。

【００２９】

図７は、本発明の第１の実施の形態において、第２のプログラム２１０のステップ２１ａの直後の、即ち初期化後のデータレコードテーブル２５１を示している。図６との違いは、エントリ数がｍではなく１（エル）であることである。

【００３０】

図８は、本発明の第１の実施の形態において、格納処理部１１３と送信部１７０がそれぞれデータレコードテーブル１５１に対する書き込みと読み出しを行っているときの、ある一時点の状態を示している。

【００３１】

矢印１５６は格納処理部１１３がエントリを書き込む方向を、矢印１５７は送信部１７０がエントリを読み出す方向を示している。すなわち、格納処理部１１３と送信部１７０とは互いに逆の順序で読み書きしている。この理由は、格納処理部１１３の書き込みと、送信部１７０の読み出しとのすれ違いを、識別情報１５２の不連続により検出するためである。以下に詳しく説明する。

【００３２】

格納処理部１１３は、エントリ９のデータレコード１５３．９、エントリ９の識別情報１５２．９、エントリ１０のデータレコード１５３．１０、エントリ１０の識別情報１５２．１０、．．．の順にデータレコードテーブル１５１のエントリを書き込んでおり、送信部１７０の読み出しとすれ違う瞬間は、エントリ１２のデータレコード１５３．１２を書き換え途中にある。送信部１７０は、エントリ１２の識別情報１５２．１２、エントリ１２のデータレコード１５３．１２、エントリ１１の識別情報１５２．１１、エントリ１１のデータレコード１５２．１１、．．．の順にデータレコードテーブル１５１のエントリを読み出している。

【００３３】

ここで、識別情報１５２．１２とデータレコード１５３．１２、および識別情

報１５２．１１に注目する。格納処理部１１３は、データレコード１５３．１２の書き込みを完了してから識別情報１５２．１２を書き込む。そのため、図８に示した時点では、識別情報１５２．１２に格納処理１１３が書き込む前の値（具体的には１２）が残っていて、この値は識別情報１５２．１１の値（具体的には６８）とは不連続になる。この時、送信部１７０は既に識別情報１５２．１２を読み込んでおり、その後識別情報１５２．１１を読み出す。そのため、上記のすれ違いが生じたときは、必ず識別番号１５２．１１と識別番号１５２．１２が不連続になる。不連続になるということはまだ、エントリ１２は書き換え中のデータレコードを含んでいることを意味する。

【００３４】

なお、もし仮に格納処理部１１３と送信部１７０とが同じ順序で読み書きした場合はこの限りではない。これを図９および図１０を用いて説明する。図９では、読み出し動作が書き込み動作に追いついて来た場合を示している。格納処理部１１３が識別番号１５２．１１を書き込んで（６１５．１）から、送信部１７０がこれを読み出す（６１５．２）。その後、送信部１７０がデータレコード１５３．１２を読み出している（６１５．３）が、このデータレコード１５３．１２は格納処理部１１３が書き込みを完了する前であるから、不正な値のまま読み出される。その後の処理は図１０に示す。

【００３５】

図１０では、まず格納処理部１１３がデータレコード１５３．１２を書き込む（６１５．４）。その後、格納処理部１１３が識別番号１５２．１２を書き込んで（６１５．５）から送信部１７０がこれを読み出し（６１５．６）ているが、この識別番号１５２．１２は正しい値（具体的には６９）が読み出される。すなわち、送信部１７０が読み出した識別番号１５２．１１と識別番号１５２．１２は、連続になる。このように、格納処理部１１３の書き込みと、送信部１７０の読み出しとの順序が同じであると、図９で示したように、６９番目のデータレコードは正しくないまま読み出しているにも係わらず識別番号は連続したものとして読み出されてしまう。従って、識別番号１５２の連続性だけでデータレコード１５３が正しく読めたことを保証できない。

【００３６】

図１１は本発明の第１の実施の形態において、図８のデータレコードテーブル１５１が、送信部１７０と受信部２７０とによって、第２の計算機ノード２０に転送された、データレコードテーブル２５１を示している。

【００３７】

図８で述べたように、エントリ９の識別情報２５２．９から、エントリ１１の識別情報２５２．１１は６６から６８と連続しており、これらに対するデータレコード２５３．９からデータレコード２５３．１１が正しく書き込まれたことを示している。さらに、データレコード２５３．１２の識別情報は１２であり連続でなく、データレコード２５３．１２が書き換え途中に読み出されたことを示している。

【００３８】

図１２は、第１の実施の形態において、データレコード参照処理部２２１の内容を示すフローチャートである。ここでは、識別番号がエントリ上で連続性があるか、どのエントリのデータレコードが不正であるかを知る。

【００３９】

まず、データレコード参照処理部２２１はカウンタ２２６を０にクリアし（ステップ２２１ａ）、ポインタ２２５を０にクリアする（ステップ２２１ｂ）。ここで、データレコード参照処理部２２１はポインタ２２５が０であるか否かの判断を行い（ステップ２２１ｃ）、該判断が真の場合は、ステップ２２１ｄへ、該判断が偽の場合は、ステップ２２１ｈへ進む。ステップ２２１ｄでは、データレコード参照処理部２２１は、カウンタ２２６が０であるか否かの判断を行い、該判断が真の場合は、ステップ２２１ｅへ、該判断が偽の場合は、ステップ２２１ｇへ進む。ステップ２２１ｅでは、データレコード参照処理部２２１は、インデクス１－１の識別情報２５２がｎ－１であるか否かの判断を行い、該判断が真の場合はステップ２２１ｆへ、該判断が偽の場合はノードＢへ進む。ステップ２２１ｆでは、データレコード参照処理部２２１は、ポインタ２２５が指すエントリの識別情報２５２が、カウンタ２２６の値と一致しているか否かを判断する。該判断が真の場合はノードＡへ進み、該判断が偽の場合はノードＢへ進む。ステッ

プ２２１ｇにおいて、データレコード参照処理部２２１は、インデクスｌ－１が指す識別情報２５２の値がカウンタ２２６の値から１を減じた数と等しいか否かを判断する。該判断が真の場合はステップ２２１ｆへ進み、該判断が偽の場合はノードＢへ進む。ステップ２２１ｈにおいて、データレコード参照処理部２２１は、カウンタ２２６が０か否かの判断を行う。該判断が真の場合、ステップ２２１ｉへ進み、該判断が偽の場合、ステップ２２１ｊへ進む。ステップ２２１ｉにおいて、データレコード参照処理部２２１は、ポインタから１を減じたインデクスの、識別情報２５２の値が、ｎ－１であるか否かの判断を行う。該判断が真の場合はステップ２２１ｆへ進み、該判断が偽の場合はノードＢへ進む。ステップ２２１ｊにおいて、データレコード参照処理部２２１は、ポインタから１を減じたインデクスの、識別情報２５２の値が、カウンタ２２６の値から１を減じた数と等しいか否かの判断を行う。該判断が真の場合はステップ２２１ｆへ進み、該判断が偽の場合はノードＢへ進む。ノードＡに進んだ場合は、データレコード参照処理部２２１は、ポインタ２２５が指すエントリのデータレコード２５３は正しいと判断し、該データレコード２５３を参照する（ステップ２２１ｋ）。さらにデータレコード参照処理部２２１は、カウンタ２２６がｎ－１より小さいか判断する。該判断が真の場合は、カウンタ２２６をインクリメントする（ステップ２２１ｎ）。該判断が偽の場合はカウンタ２２６を０にクリアする（ステップ２２１ｏ）。さらに、データレコード参照処理部２２１は、ポインタ２２５がｌ－１より小さいか否かを判断する。該判断が真の場合は、データレコード参照処理部２２１は、ポインタ２２５をインクリメントして（ステップ２２１ｑ）、ステップ２２１ｃへ進む。該判断が偽の場合は、ステップ２２１ｂへ進む。ノードＢへ進んだ場合は、データレコード参照処理部２２１は、ポインタ２２５が指すエントリのデータレコード２５３は不正と判断し、該データレコード２５３は参照しない（ステップ２２１ｌ）。さらに、データレコード参照処理部２２１は、ポインタ２２５、カウンタ２２６の値を変更せずに、ステップ２２１ｃから処理を繰り返す。

【００４０】

ここで、データレコードが不正と判断された場合はポインタはインクリメント

されない。従って、次のデータの転送はこのポインタで表されるエントリを含むいくつかのエントリのデータレコードが適当な時間間隔後に転送されることになる。

【００４１】

本発明の第１の実施形態ではＣＰＵの負荷の削減効果が大きい。

【００４２】

次に、図１３から図１８を用いて、第２の本発明の実施の形態と第１の本発明の実施の形態との違いを説明する。

図１３は、第２の本発明の実施の形態の全体構成図である。図１との違いは、第1のプログラム１１０が誤り検出符号生成処理部１１４を含むことと、第２のプログラム２１０が誤り検出符号検査処理部２２２を含むこと、データレコードテーブル１５１に誤り検査符号１５４を含むこと、データレコードテーブル２５１に誤り検査符号２５４を含むことである。

【００４３】

誤り検出符号生成処理部１１４は、データレコードエントリの識別情報１５２と、データレコード１５３の組から、誤り検出符号１５４を生成する。誤り検出符号検査処理部２２２は、誤り検出符号２５４が、データレコードエントリの識別情報２５２と、データレコード２５３との組から生成された符号（誤りなし）か否（誤りあり）かを検査する。ここで、誤り検出符号を採用した理由を簡単に説明しておく。第１の実施例ではデータレコードテーブル１５１にジャーナルデータなどのデータレコードを書き込む方向と反対の方向に転送のための読み出しを進めることにより不正なデータを検出する方法を採ったのに対して、同一方向に読み出す場合を想定している。そして、データレコードの正しさを誤り検出符号を利用して保証しようとしたものである。

【００４４】

図１４は、第２の本発明の実施の形態において、第１のプログラム１１０の処理を示すフローチャートである。

図２との違いは、ステップ１１ｄおよびステップ１１ｅ、ステップ１１ｆを含むこと、ステップｇを含まないことである。ステップ１１ｄでは、格納処理部１

１３がステップ１１ａのデータレコード１５３とステップ１１ｃの識別情報１５２とを組にして、誤り検出符号生成処理部１１４に渡す。ステップ１１ｅでは、誤り検出符号生成処理部１１４が、ステップ１１ｄで渡された情報から誤り検出符号１５４を生成し、格納処理部１１３に返す。ステップ１１ｆでは、格納処理部１１３が、ステップ１１ａのデータレコード１５３とステップ１１ｃの識別情報１５２とステップ１１ｅの誤り検出符号１５４とを組にして、ポインタ１１５が指すエントリに格納する。

【００４５】

図１５は、第２の本発明の実施の形態において、格納処理部１１３の内容を示すフローチャートである。

図５との違いは、ステップ１１３ｉとステップ１１３ｊを含むことである。ステップ１１３ｉにおいて、格納処理部１１３は、図１４で示したステップ１１ｄを行い、誤り検出符号生成処理部１１４から誤り検出符号１５４を得る。ステップ１１３ｊにおいて、格納処理部１１３は、ステップ１１３ｉで取得した誤り検出符号１５４をポインタ１１５が指すエントリに格納する。

【００４６】

図１６は、本発明の第２の実施の形態において、初期化後のデータレコードテーブル１５１を示している。

図６との違いは、誤り検出符号１５４を含むこと、データレコード１５３の初期化を必ず行う必要があることである。何故ならば古いデータではあるがデータレコードと誤り検出符号とが整合したデータが残っているとそれが新しく格納された正しいデータレコードなのか、古い（正しくない）データなのかの区別が付かなくなるからである。格納処理部１１３は、ステップ１１３ａにおいて、これらの誤り検出符号１５４を、識別情報１５２、データレコード１５３から生成した誤り訂正符号以外の値、即ち、不正な符号を格納する。

【００４７】

図１７は、本発明の第２の本発明の実施の形態において、データレコードテーブル１５１の、ある一時点の状態を示している。

図８との違いは、誤り検出符号１５４を含むことである。図１７で、格納処理

部１１３がエントリ１２のデータレコード１５３に書き込んでいる途中で、送信部１７０が当該エントリを読み出しているとする。このとき、エントリ１２の誤り検出符号１５４．１２は、当該エントリから生成された誤り検出符号でないため、不正な符号となる。図１７のその他の誤り検出符号１５４．９から誤り検出符号１５４．１１はそれぞれのエントリから生成された誤り検出符号であるため、正しい符号である。

【００４８】

図１８は、本発明の第２の実施の形態において、データレコード参照処理部２２１の内容を示すフローチャートである。ここでは誤り検出符号によって読み出されたデータレコードが正しいものかどうかを判定している。

【００４９】

図１２との違いは、ステップ２２１ｒおよび２２１ｓを含むこと、ステップ２２１ｃからステップ２２１ｅ、およびステップ２２１ｇからステップ２２１ｊを含まないことである。ステップ２２１ｒにおいて、データレコード参照処理部２２１はポインタ２２５が指すエントリを誤り検出符号検査処理部２２２に渡す。ステップ２２１ｓにおいて、データレコード参照処理部２２１は、誤り検出符号検査処理部２２２の結果から、当該レコードに誤りが含まれたか否かを判断する。該判断が真の場合はノードＢへ進み、該判断が偽の場合はノードＡに進む。

【００５０】

本発明の第２の実施の形態においては、読み取り順序が同一方向のものでもデータレコードの正しさが保証できる。

【００５１】

次に、図１９から図２２を用いて、第３の本発明の実施の形態と第１の本発明の実施の形態との違いを説明する。

図１９は、第３の本発明の実施の形態の全体構成図である。図１との違いは、第1のプログラム１１０がデータ送信要求生成処理部１２２を含むこと、第２のプログラム２１０がタイマ２１１とデータ受信要求生成処理部２１２を含まないこと、である。データ送信要求生成処理部１２２は送信部１７０に対し、データレコードテーブル１５１の送信要求を生成する。即ち、ＲＤＭＡ－Ｗｒｉｔｅを

使って送信側の主導によりデータの転送を行なう場合を示している。このとき、第１の計算機ノードには図１の構成に比べて負荷はかかるが同期を取らない転送方式としているため従来例よりも負荷は小さいものとなっている。データ送信要求生成処理部１２２は送信データがある程度たまったときに送信部１７０にデータの読み出しを行なわせるものである。データの転送は第２のプログラムがポーリングして転送データがあることを認知して実施される。

【００５２】

図２０は、本発明の第３の実施の形態における、第１のプログラム１１０の処理を示すフローチャートである。

図２との違いは、ステップ１１ｉからステップ１１１を含むことである。ステップ１１ｉにおいて、第１のプログラム１１０はデータレコードテーブル１５１を送信するか否かの判断をする。即ち、本発明の第３の実施の形態おいて、データ送信の間隔は任意である。リアルタイム性を向上する為に望ましくは、該時間間隔を短くする、即ち、なるべく多くの場合についてステップ１１ｉにおける判断を真とする。データ送信の時間間隔の調整はデータ送信要求生成処理部１２２がステップ１１ｉで行なう。

【００５３】

データ転送効率を向上する為に望ましくは、最後にデータ送信を行った時点から現在までに、格納処理部１１３がデータレコードテーブル１５１の半分に相当するエントリを格納したときにステップ１１ｉにおける判断を真とする。

【００５４】

該判断が真の場合、ステップ１１ｊに進み、偽の場合は１１ａに進む。ステップ１１ｊにおいては、データ送信要求生成処理部１２２が、データ送信要求を作成し、送信部１７０を起動する。さらに、ステップ１１ｋとステップ１１１において、第１のプログラム１１０は、ステップ１１ｊで起動した送信が完了したことを待つ。その後、第１のプログラム１１０はステップ１１ａから処理を続行する。本発明の第３の実施の形態では、第２のプログラム２１０は、図３に示すステップ２１ｆを実行するのみである。

【００５５】

本発明の第３の実施の形態では、格納処理部１１３は、図５に示す処理と同一だが、ステップ１１２ｇとステップ１１３ｈの順序の入れ替えが可能であることが異なる。本発明の第３の実施の形態では、データレコードテーブル１５１の読み書き順序について、図８に示す順序でなくても構わない。

【００５６】

図２１は、本発明の第３の実施の形態において、データレコードテーブル１５１が、送信部１７０と受信部２７０とによって、第２の計算機ノード２０に転送された、データレコードテーブル２５１を示している。

【００５７】

矢印２５６は受信部２７０がエントリを書き込む方向を、矢印２５７は第２のプログラム２１０がエントリを読み出す方向を示している。データレコード２５３が正しく読めたか否かの判定は、本発明の第１の実施の形態と同様である。

【００５８】

図２２は、第３の実施の形態において、データレコード参照処理部２２１の内容を示すフローチャートである。

図２２は図１２と異なり、ステップ２１１ｕからステップ２１１ｘを含み、ステップ２１１ｍからステップ２１１ｑを含まず、ステップ２１１ｆをステップ２１ｃの前に実行する。即ち、本発明の第３の実施の形態によれば、データレコード参照処理部２２１はインデクスの大きいほうの識別情報２５２から参照する。ステップ２１１ｕでは、データレコード参照処理部２１１はポインタ２２５、カウンタ２２６それぞれに、同じ任意の自然数を加える。ここで望ましくは、該自然数は以降のステップ２２１ｆの判断が真となる、最大の数とする。ただし、この最適値を予測することは困難なので、例えば前回のステップ２２１ｕにおいて加算した自然数を記憶しておき、今回のステップ２１１ｕでは、該自然数から、前回のステップ２２１ｕ処理から不正レコードを読み出した数を差し引いた数、またはそれに近い数を指定する。

【００５９】

さらに、ステップ２１１ｖでは、データレコード参照処理部２１１はポインタ

２２５、カウンタ２２６のラップ処理を行う。すなわち、ポインタ２２５が０以上１－１以下になるまで、ポインタ２２５から１を繰り返し除し、カウンタ２２６が０以上ｎ－１以下になるまで、カウンタ２２６からｎを繰り返し除す。ステップ２１１ｆの判断が真の場合、データレコード参照処理部２１１はステップ２１１ｃへ進み、偽の場合はノードＢへ進む。ステップ２２１Ｗにおいて、データレコード参照処理部２１１はポインタ２２５から１を減じた値が指すエントリ（ポインタ２２５が０の場合は、エントリ１－１）が既に正しく読み込み済みか判断する。

【００６０】

該判断が真の場合、データレコード参照処理部２１１はステップ２２１ｕに進み、該判断が偽の場合、ステップ２１１ｘに進む。ステップ２１１ｘにおいて、データレコード参照処理部２１１は、カウンタ２２６、ポインタ２２５をそれぞれデクリメントする。本発明の第３の実施の形態ではＲＤＭＡ－Ｒｅａｄをサポートしていないものにも適用できる。

【００６１】

次に、図２３から図２４を用いて、第４の本発明の実施の形態と第２の本発明の実施の形態との違いを説明する。

図２３は、第４の本発明の実施の形態の全体構成図である。図２３は、図１３と異なり、第1のプログラム１１０がデータ送信要求生成処理部１２２を含み、第２のプログラム２１０がタイマ２１１とデータ受信要求生成処理部２１２を含まない。データ送信要求生成処理部１２２は本発明の第３の実施の形態と変わらない。

【００６２】

図２４は、第４の本発明の実施の形態における、第1のプログラム１１０の処理を示すフローチャートである。

図２４は、図１４と異なり、ステップ１１ｉからステップ１１１を含む。これらのステップは、本発明の第３の実施の形態と同様である。本発明の第４の実施の形態では読み取り順序が同一で、ＲＤＭＡ－Ｒｅａｄをサポートしていないものにも適用出来る。

【００６３】

次に、図２５から図２７を用いて、第５の本発明の実施の形態と第１の本発明の実施の形態との違いを説明する。図２５は、第５の本発明の実施の形態の全体構成図である。図２５は図１と異なり、第１のプログラム１１０に通知処理部１１７を含み、第２のプログラム２１０にタイマ２１１を含まない。通知処理部１１７はデータレコードがある程度たまったとき送信部１７０に読み出しをなさせるとともに第２のプログラムに読み取りのきっかけを与えるものでＲＤＭＡ－Ｗｒｉｔｅの割り込みをするものである。

【００６４】

図２６は、第５の本発明の実施の形態における、第1のプログラム１１０の処理を示すフローチャートである。

図２６は、図２と異なり、ステップ１１ｍを有する。ステップ１１ｍにおいて、通知処理部１１７は、送信部１７０に、第２のプログラム２１０が受信するイベント通知を依頼する。本実施例では第２のプログラムにタイマが不要で送信側のトリガでデータの転送が制御される。つまり、データの転送の時間間隔は通知処理部１１７の制御による。

【００６５】

図２７は、第５の実施の形態において、第２のプログラム２１０の処理を示すフローチャートである。

図２７は、図３と異なり、ステップ２１ｈを含み、ステップ２１ｂおよびステップ２１ｇを含まない。ステップ２１ｂにおいて、第２のプログラム２１０は通知処理部１１７からのイベント通知を待つ。本発明の第５の実施の形態ではＣＰＵの負荷の削減効果が大きく、送信側の通知をトリガとしてデータの転送が行なわれるので受信側の負担が小さい。

【００６６】

次に、図２８を用いて、第６の本発明の実施の形態と第２の本発明の実施の形態との違いを説明する。

図２８は、第６の本発明の実施の形態の全体構成図である。

【００６７】

図２８は図１３と異なり、第１のプログラム１１０に通知処理部１１７を含み、第２のプログラム２１０にタイマ２１１を含まない。その他の、第６の本発明の実施の形態と第２の本発明の実施の形態との違いは、第５の本発明の実施の形態と第１の本発明の実施の形態との違いと同一なので説明を省略する。これはデータレコードテーブル１５１へのジャーナルデータなどのデータレコードを書き込む方向とその読み出し方向とが同じである場合を想定し、且つ、送信側でデータの転送時間間隔を制御する場合である。本発明の第６の実施の形態では読み取り順序が同一のものでもサポートでき、ＣＰＵ負荷の削減の効果が大きい。また、送信側の通知をトリガとしているので受信側の負担が小さい。

【００６８】

次に、図２９を用いて、第７の本発明の実施の形態と第３の本発明の実施の形態との違いを説明する。

図２９は、第７の本発明の実施の形態の全体構成図である。図２９は図１９と異なり、第１のプログラム１１０にデータ送信要求生成・通知処理部１２３を含み、第1のプログラム１１０にデータ送信要求生成処理部１２２を含まない。

【００６９】

データ送信要求生成・通知処理部１２３は、データ送信要求生成処理部１２２と異なり、送信部１１７を起動するとき、データ送信要求と通知要求を組みにして送信部に渡す。ただし、本発明を実施するためには、データ送信要求と通知要求は必ずしも組みにする必要はない。本発明の第７の実施の形態ではＣＰＵの負荷の削減効果が大きく、ＲＤＭＡ－Ｒｅａｄをサポートしていないものにも適用が可能である。また、送信側の通知によるため受信側の負担は小さい。

【００７０】

次に、図３０を用いて、第８の本発明の実施の形態と第４の本発明の実施の形態との違いを説明する。

図３０は、第８の本発明の実施の形態の全体構成図である。図３０は図２３と異なり、第１のプログラム１１０にデータ送信要求生成・通知処理部１２３を含み、第1のプログラム１１０にデータ送信要求生成処理部１２２を含まない。本

発明の第８の実施の形態では読み取りが同一方向でも適用でき、ＲＤＭＡ－Ｒｅａｄをサポートしていないものにも適用出来る。また、送信側の通知によるため受信側の負荷は小さい。

【００７１】

次に、図３１を用いて、第９の本発明の実施の形態を説明する。

図３１は、第９の本発明の実施の形態の全体構成図である。第1の計算機ノード１０は基幹系システム５１０を含み、第2の計算機ノード２０は情報系システム２０を含む。第1のプログラム１１０は、第９の本発明の実施の形態ではオンライン・トランザクション・プロセッシング（ＯＬＴＰ）である。データレコード１５３は第９の本発明の実施の形態では、ＯＬＴＰ１１０が処理過程を保存するために出力するジャーナルである。第２のプログラム２１０は、第９の本発明の実施の形態ではデータベース管理システム（ＤＢＭＳ）である。

【００７２】

第９の本発明の実施の形態では、第1の本発明の実施の形態から、第８の本発明の実施の形態に示した処理内容の具体的応用を示すもので、これらのいずれか一つと同様の処理を行って、ＯＬＴＰ１１０が出力するジャーナル１５３を情報系システム５２０に転送する。

【００７３】

【発明の効果】

本発明により、データを送信するプログラムと受信するプログラムとの同期オーバヘッドが低減する。

【図面の簡単な説明】

【図１】本発明の第１の実施の形態の全体構成図。

【図２】第１のプログラム１１０の処理を示すフローチャート。

【図３】第２のプログラム２１０の処理を示すフローチャート。

【図４】識別情報出力処理１１２の処理を示すフローチャート。

【図５】格納処理１１３を示すフローチャート。

【図６】データレコードテーブル１５１を示す図。

【図７】データレコードテーブル２５１を示す図。

【図８】データレコードテーブル１５１を示す図。

【図９】同方向に読み出しと書き込みを行なったときのデータレコードテーブルの状態を説明する図（１）。

【図１０】同方向に読み出しと書き込みを行なったときのデータレコードテーブルの状態を説明する図（２）。

【図１１】データレコードテーブル２５１を示す図。

【図１２】データレコード参照処理２２１の内容を示すフローチャート。

【図１３】第２の本発明の実施の形態の全体構成図。

【図１４】第１のプログラム１１０の処理を示すフローチャート。

【図１５】格納処理１１３の内容を示すフローチャート。

【図１６】データレコードテーブル１５１を示す図。

【図１７】データレコードテーブル１５１を示す図。

【図１８】データレコード参照処理２２１の内容を示すフローチャート。

【図１９】第３の本発明の実施の形態の全体構成図。

【図２０】第１のプログラム１１０の処理を示すフローチャート。

【図２１】データレコードテーブル２５１を示す図。

【図２２】データレコード参照処理２２１の内容を示すフローチャート。

【図２３】第４の本発明の実施の形態の全体構成図。

【図２４】第１のプログラム１１０の処理を示すフローチャート。

【図２５】第５の本発明の実施の形態の全体構成図。

【図２６】第１のプログラム１１０の処理を示すフローチャート。

【図２７】第２のプログラム２１０の処理を示すフローチャート。

【図２８】第６の本発明の実施の形態の全体構成図。

【図２９】第７の本発明の実施の形態の全体構成図。

【図３０】第８の本発明の実施の形態の全体構成図。

【図３１】第９の本発明の実施の形態の全体構成図。

【符号の説明】

１０：第１の計算機ノード

２０：第２の計算機ノード

１１０：第１のプログラム

１１１：データレコード出力処理部

１１２：識別情報出力処理部

１１３：格納処理部

１１５：ポインタ

１１６：カウンタ

１５０：主記憶

１５１：データレコードテーブル

１５２：識別情報

１５３：データレコード

１７０：送信部

２１０：第２のプログラム

２１１：タイマ

２１２：データ受信要求生成処理部

２２１：データレコード参照処理部

２２５：ポインタ

２２６：カウンタ

２５０：主記憶

２５１：データレコードテーブル

２５２：識別情報

２５３：データレコード

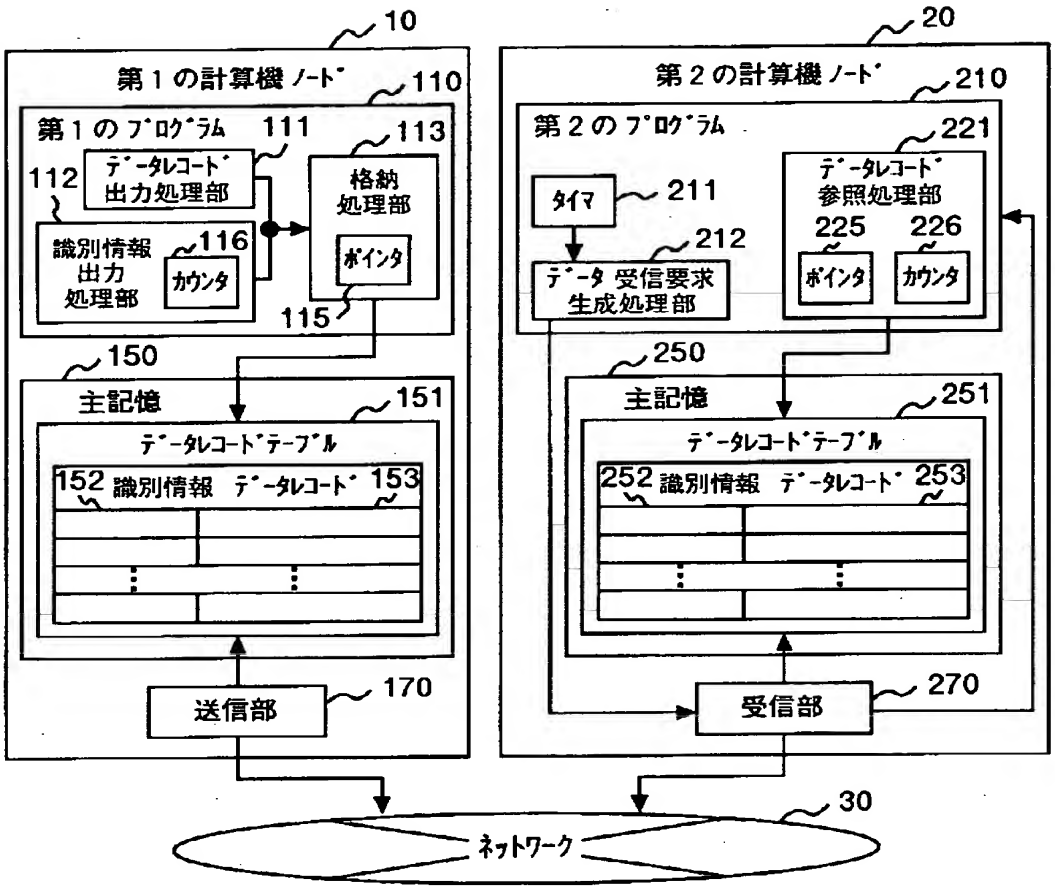
２７０：受信部

３０：ネットワーク

【書類名】　図面

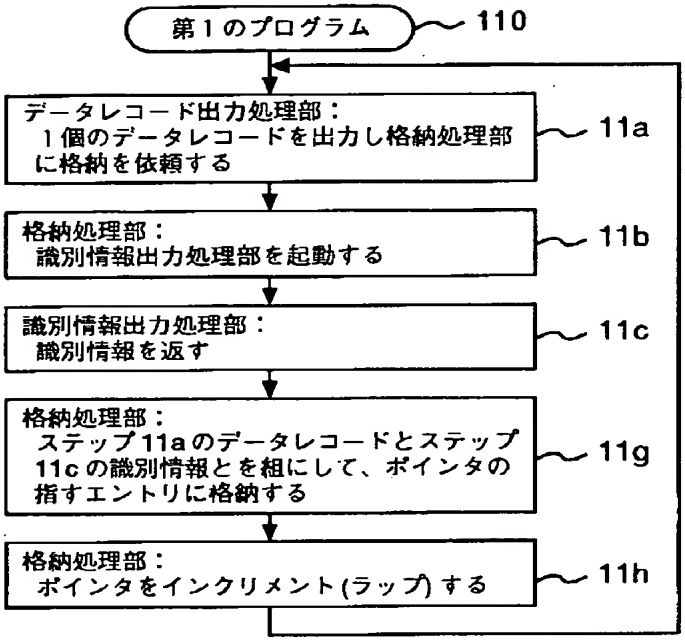
【図１】

図　1


10
第１の計算機ノード
110
第１のプログラム
111
113
112
データレコード出力処理部
格納処理部
識別情報出力処理部
116
カウンタ
ポインタ
115
150
主記憶
151
データレコードテーブル
152 識別情報
データレコード
153
送信部
170
20
第２の計算機ノード
210
第２のプログラム
221
データレコード参照処理部
タイマ
211
212
データ受信要求生成処理部
225
226
ポインタ
カウンタ
250
主記憶
251
データレコードテーブル
252 識別情報
データレコード
253
受信部
270
30
ネットワーク

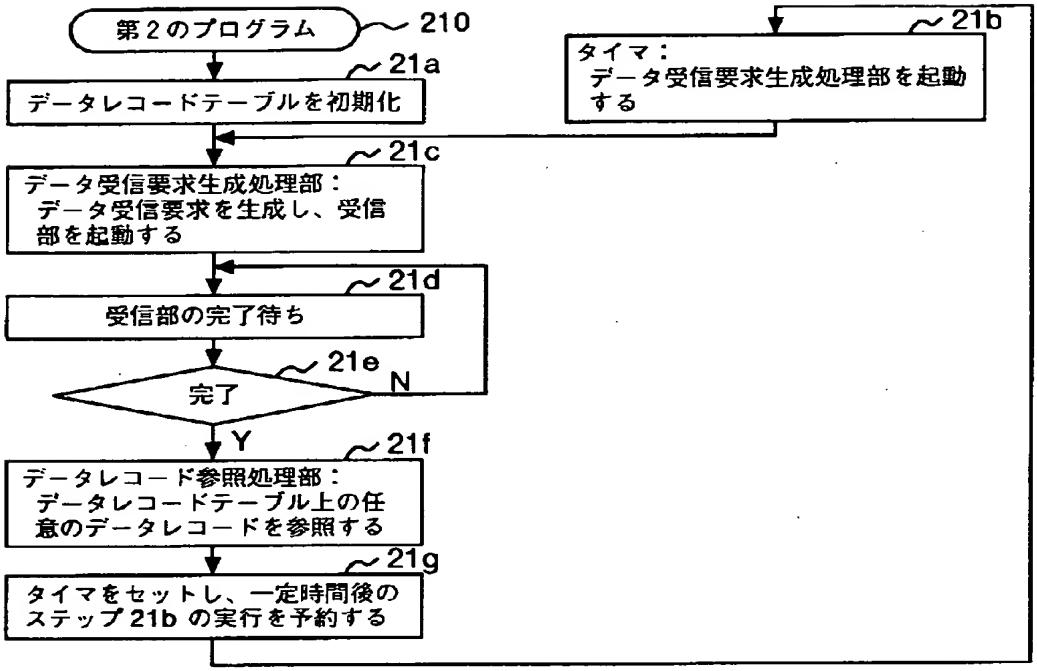
【図２】

図　2


第１のプログラム
110
データレコード出力処理部：
１個のデータレコードを出力し格納処理部に格納を依頼する
11a
格納処理部：
識別情報出力処理部を起動する
11b
識別情報出力処理部：
識別情報を返す
11c
格納処理部：
ステップ11aのデータレコードとステップ11cの識別情報とを組にして、ポインタの指すエントリに格納する
11g
格納処理部：
ポインタをインクリメント(ラップ)する
11h

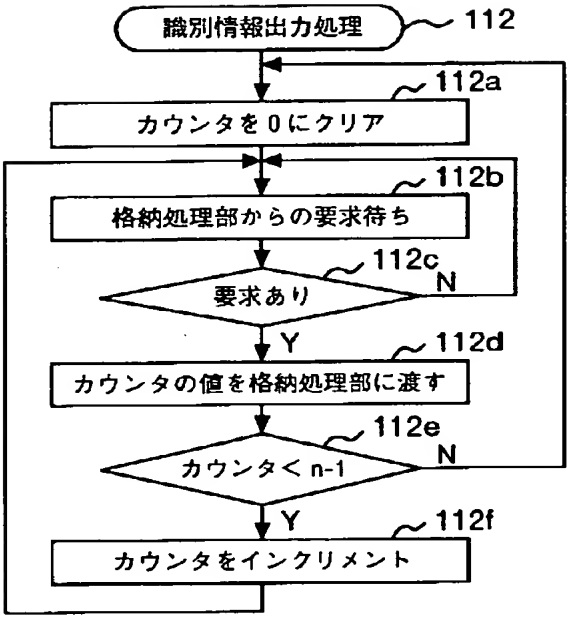
【図３】

図　3


第２のプログラム
210
21a
データレコードテーブルを初期化
21b
タイマ：
データ受信要求生成処理部を起動する
21c
データ受信要求生成処理部：
データ受信要求を生成し、受信部を起動する
21d
受信部の完了待ち
21e
完了
N
Y
21f
データレコード参照処理部：
データレコードテーブル上の任意のデータレコードを参照する
21g
タイマをセットし、一定時間後のステップ21b の実行を予約する

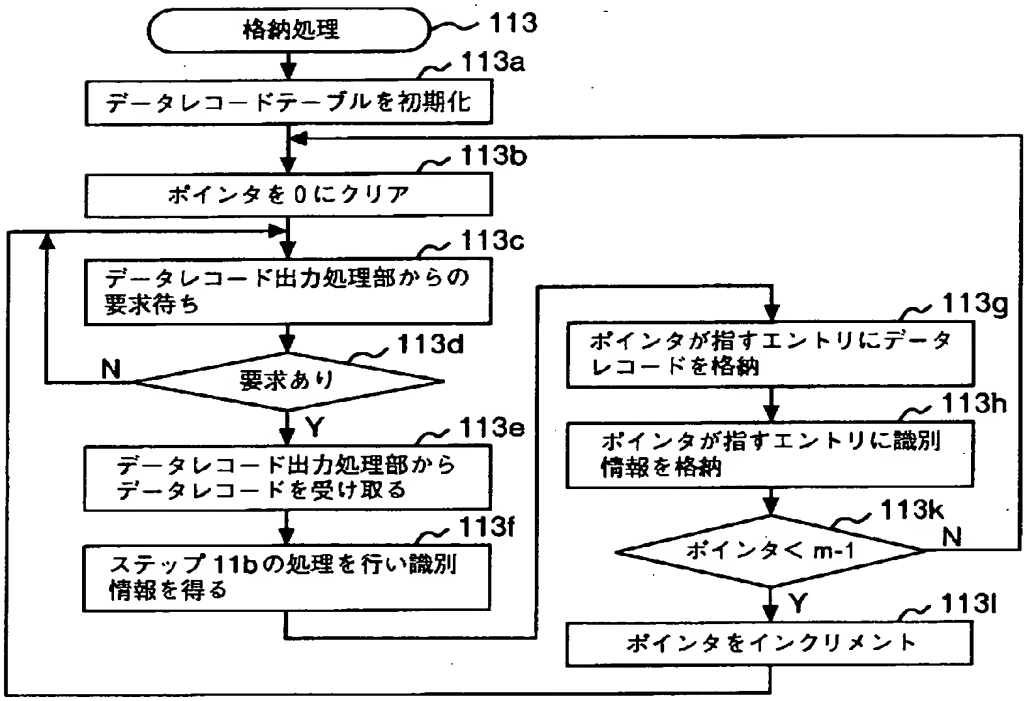
【図４】

図　4


識別情報出力処理
112
112a
カウンタを０にクリア
112b
格納処理部からの要求待ち
112c
要求あり
N
Y
112d
カウンタの値を格納処理部に渡す
112e
カウンタ＜ n-1
N
Y
112f
カウンタをインクリメント

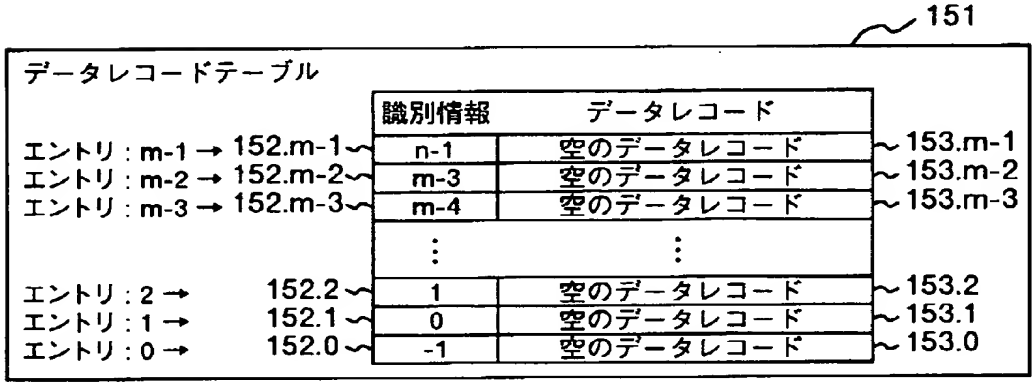
【図５】

図　5


格納処理
113
113a
データレコードテーブルを初期化
113b
ポインタを０にクリア
113c
データレコード出力処理部からの要求待ち
113d
N
要求あり
Y
113e
データレコード出力処理部からデータレコードを受け取る
113f
ステップ11bの処理を行い識別情報を得る
113g
ポインタが指すエントリにデータレコードを格納
113h
ポインタが指すエントリに識別情報を格納
113k
ポインタ＜m-1
N
Y
113l
ポインタをインクリメント

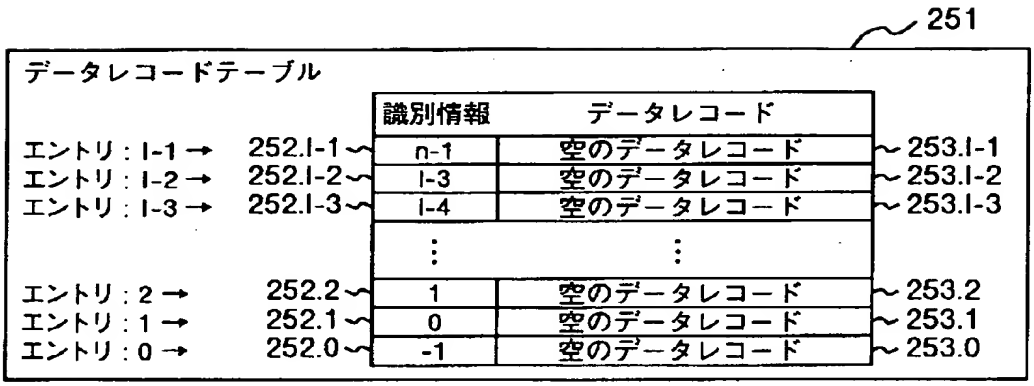
【図６】

図　6


151
データレコードテーブル
識別情報
データレコード
エントリ：m-1 → 152.m-1
n-1
空のデータレコード
153.m-1
エントリ：m-2 → 152.m-2
m-3
空のデータレコード
153.m-2
エントリ：m-3 → 152.m-3
m-4
空のデータレコード
153.m-3
エントリ：2 → 152.2
1
空のデータレコード
153.2
エントリ：1 → 152.1
0
空のデータレコード
153.1
エントリ：0 → 152.0
-1
空のデータレコード
153.0

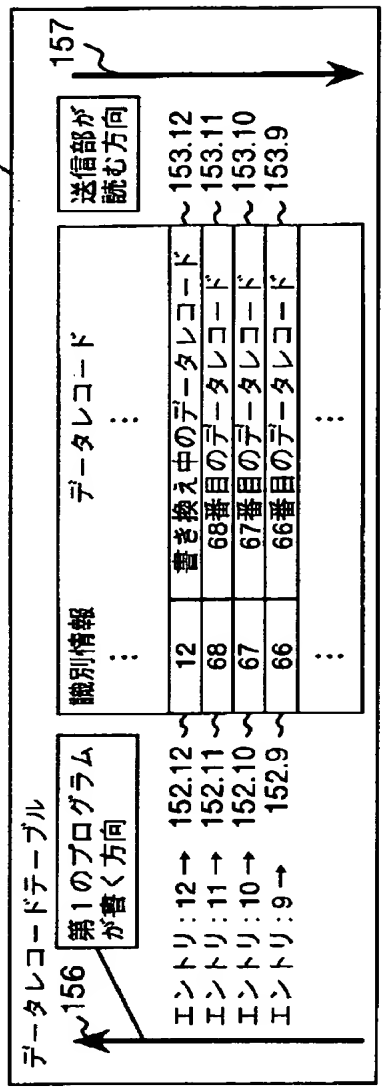
【図７】

図　7


251
データレコードテーブル
識別情報
データレコード
エントリ：I-1 →
252.I-1
n-1
空のデータレコード
253.I-1
エントリ：I-2 →
252.I-2
I-3
空のデータレコード
253.I-2
エントリ：I-3 →
252.I-3
I-4
空のデータレコード
253.I-3
エントリ：2 →
252.2
1
空のデータレコード
253.2
エントリ：1 →
252.1
0
空のデータレコード
253.1
エントリ：0 →
252.0
-1
空のデータレコード
253.0

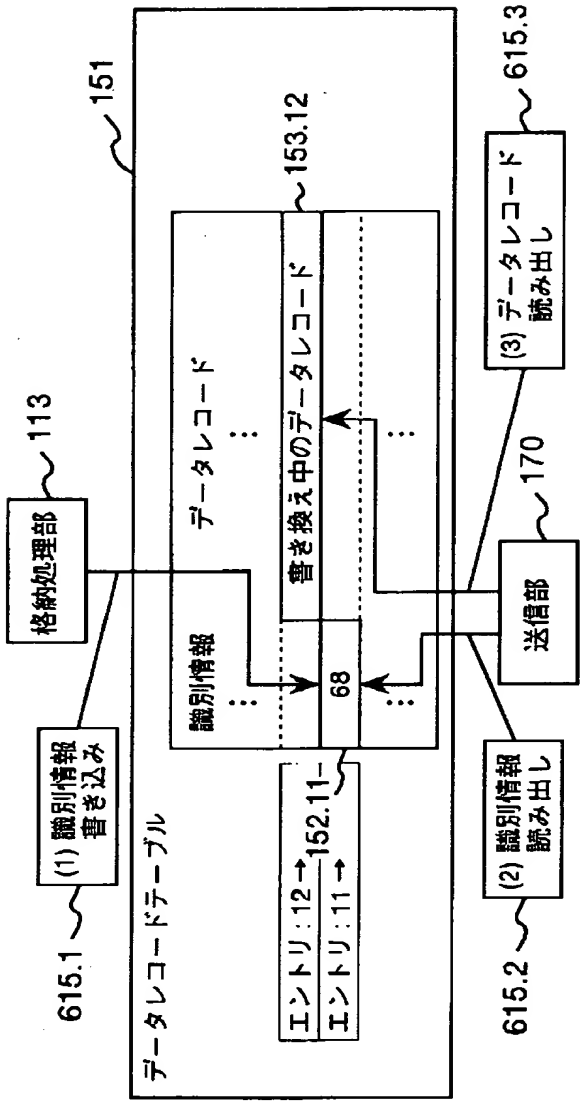
【図８】

図 8


151
157
データレコードテーブル
156
第１のプログラムが書く方向
送信部が読む方向
識別情報
データレコード
12
68
67
66
書き換え中のデータレコード
68番目のデータレコード
67番目のデータレコード
66番目のデータレコード
153.12
153.11
153.10
153.9
エントリ：12→ 152.12
エントリ：11→ 152.11
エントリ：10→ 152.10
エントリ：9→ 152.9

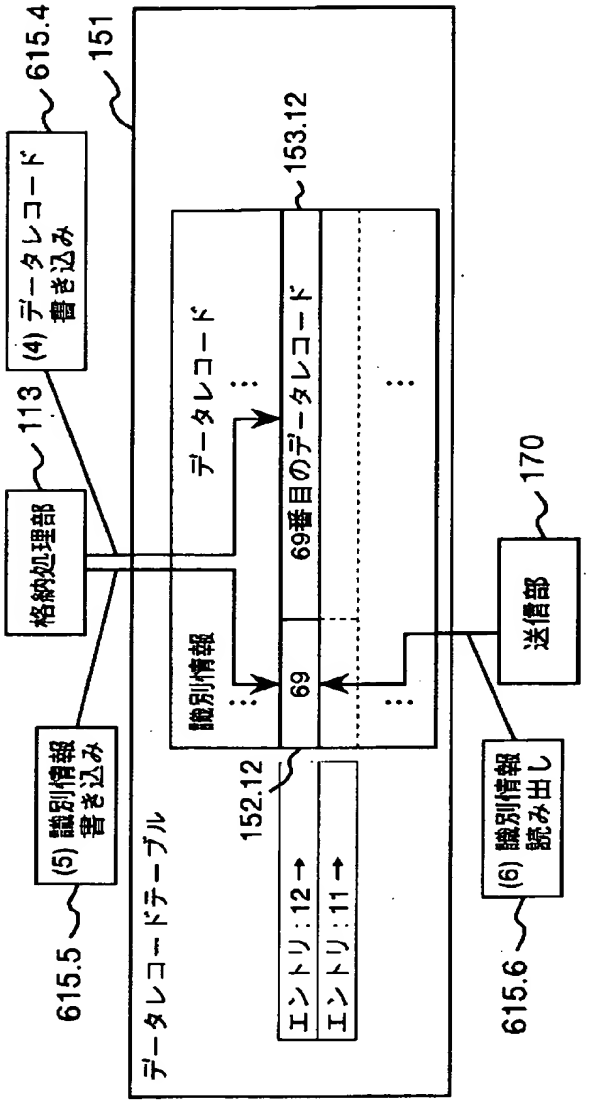
【図９】

図 9


151
153.12
615.3
(3) データレコード 読み出し
113
格納処理部
データレコード
書き換え中のデータレコード
170
送信部
識別情報
68
(1) 識別情報 書き込み
エントリ:12→
エントリ:11→
152.11
615.1
データレコードテーブル
(2) 識別情報 読み出し
615.2

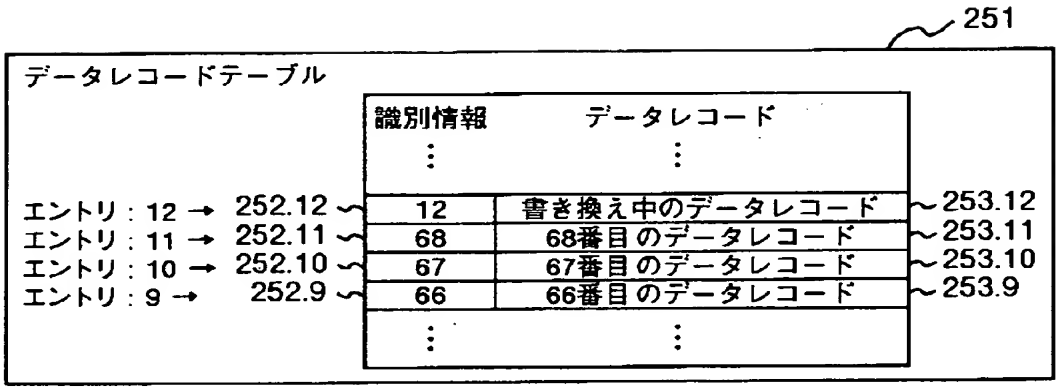
【図１０】

図　１０


615.4
151
(4) データレコード書き込み
153.12
113
格納処理部
データレコード
69番目のデータレコード
170
識別情報
69
送信部
(5) 識別情報書き込み
152.12
(6) 識別情報読み出し
615.5
エントリ：12→
エントリ：11→
615.6
データレコードテーブル

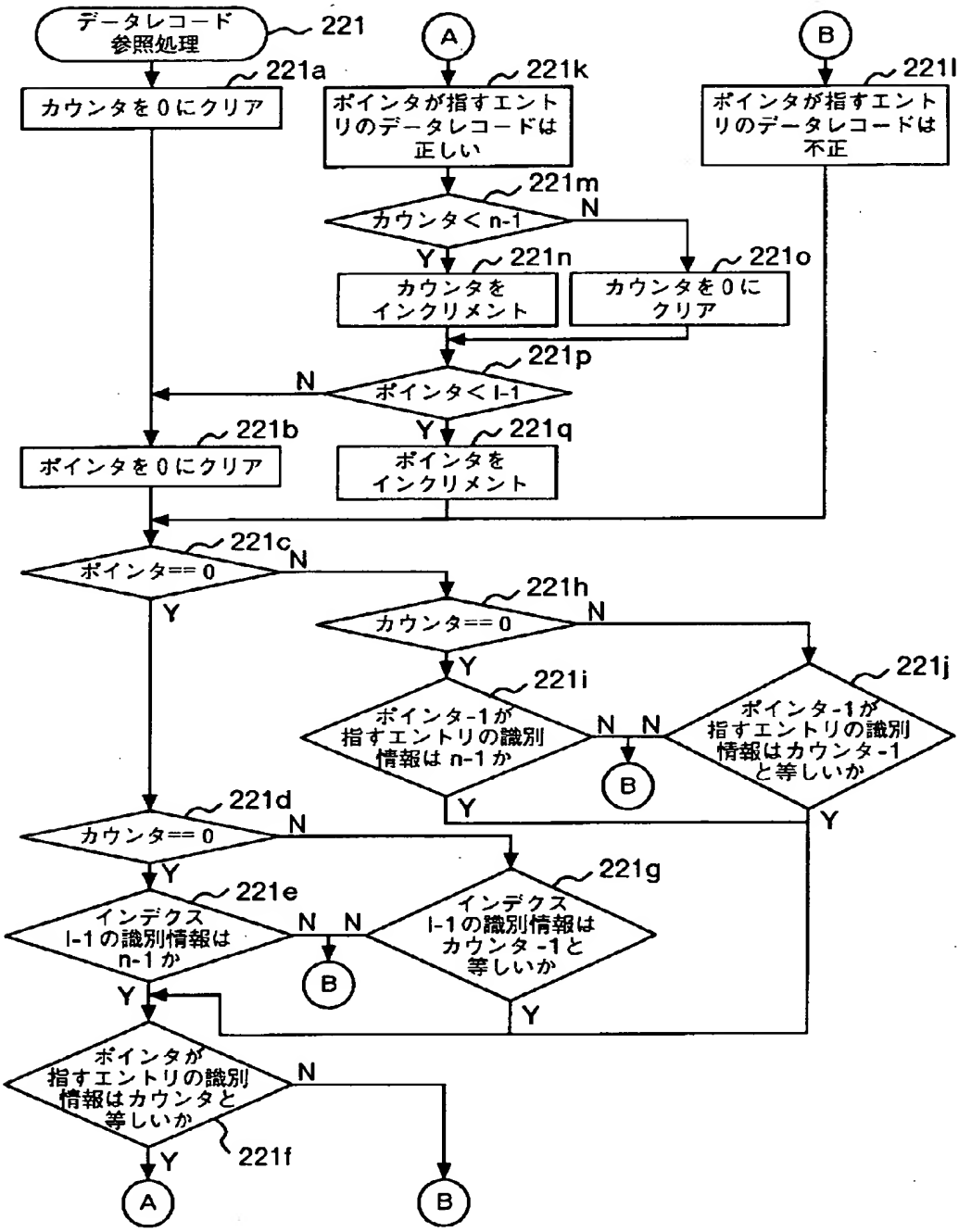
【図１１】

図　１１


251
データレコードテーブル
識別情報
データレコード
エントリ：12 → 252.12
エントリ：11 → 252.11
エントリ：10 → 252.10
エントリ：9 → 252.9
12
書き換え中のデータレコード
253.12
68
68番目のデータレコード
253.11
67
67番目のデータレコード
253.10
66
66番目のデータレコード
253.9

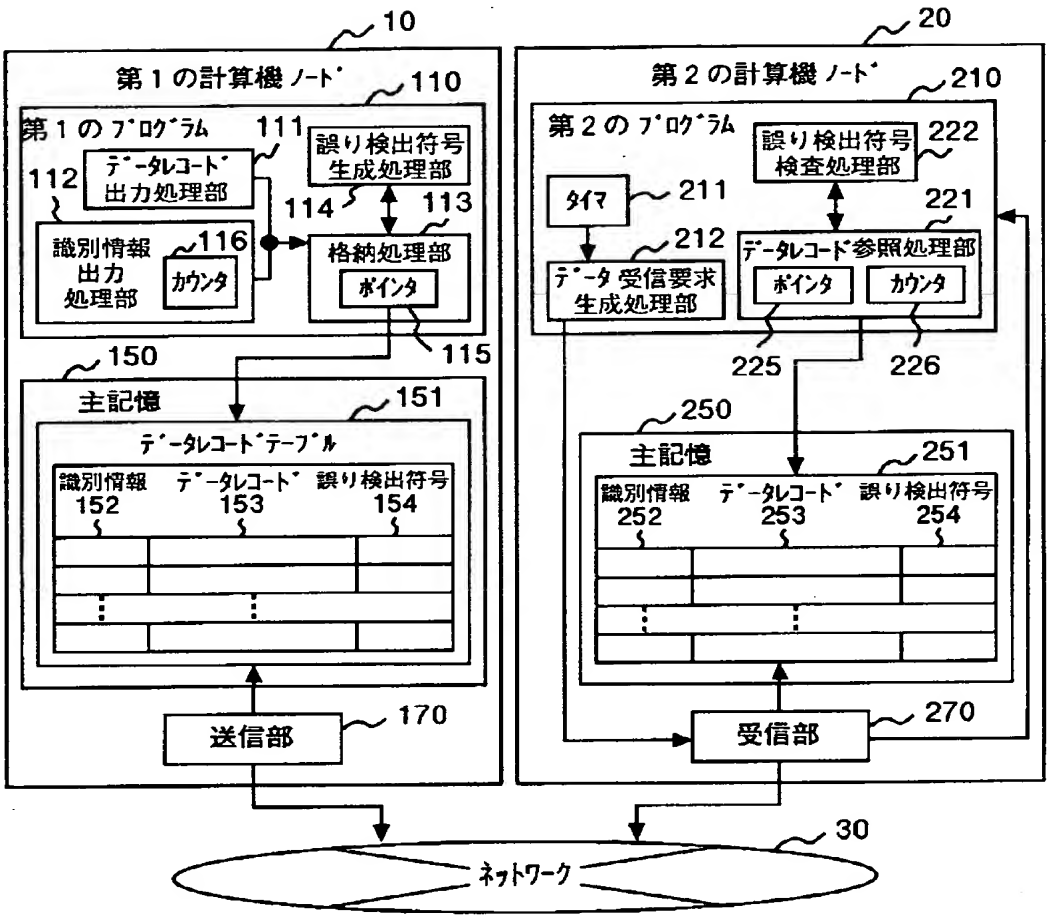
【図１２】

図　１２


データレコード参照処理
221
A
B
221a
カウンタを０にクリア
221k
ポインタが指すエントリのデータレコードは正しい
221l
ポインタが指すエントリのデータレコードは不正
221m
カウンタ＜n-1
N
Y
221n
カウンタをインクリメント
221o
カウンタを０にクリア
221p
ポインタ＜l-1
N
Y
221b
ポインタを０にクリア
221q
ポインタをインクリメント
221c
ポインタ==０
N
Y
221h
カウンタ==０
N
Y
221i
ポインタ-1が指すエントリの識別情報はn-1か
N
N
B
Y
221j
ポインタ-1が指すエントリの識別情報はカウンタ-1と等しいか
Y
221d
カウンタ==０
N
Y
221e
インデクスl-1の識別情報はn-1か
N
N
B
Y
221g
インデクスl-1の識別情報はカウンタ-1と等しいか
Y
ポインタが指すエントリの識別情報はカウンタと等しいか
N
Y
221f
A
B

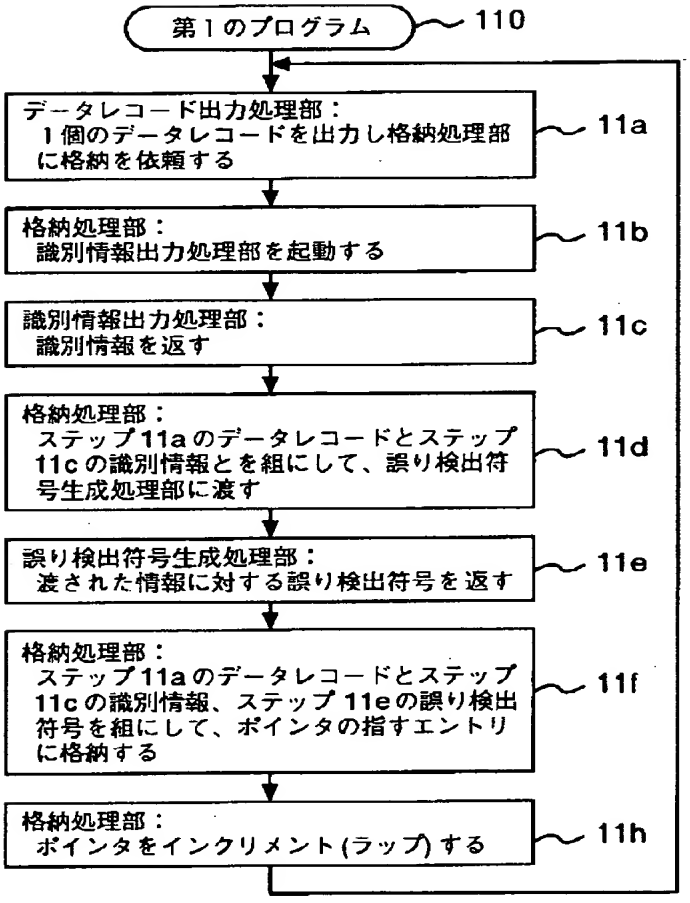
【図１３】

図　１３


10
第１の計算機ノード
110
第１のプログラム
111
誤り検出符号生成処理部
112
データレコード出力処理部
114
113
識別情報出力処理部
116
カウンタ
格納処理部
ポインタ
115
150
主記憶
151
データレコードテーブル
識別情報 152
データレコード 153
誤り検出符号 154
送信部
170
20
第２の計算機ノード
210
第２のプログラム
誤り検出符号検査処理部
222
タイマ
211
221
データ受信要求生成処理部
212
データレコード参照処理部
ポインタ
カウンタ
225
226
250
主記憶
251
識別情報 252
データレコード 253
誤り検出符号 254
受信部
270
30
ネットワーク

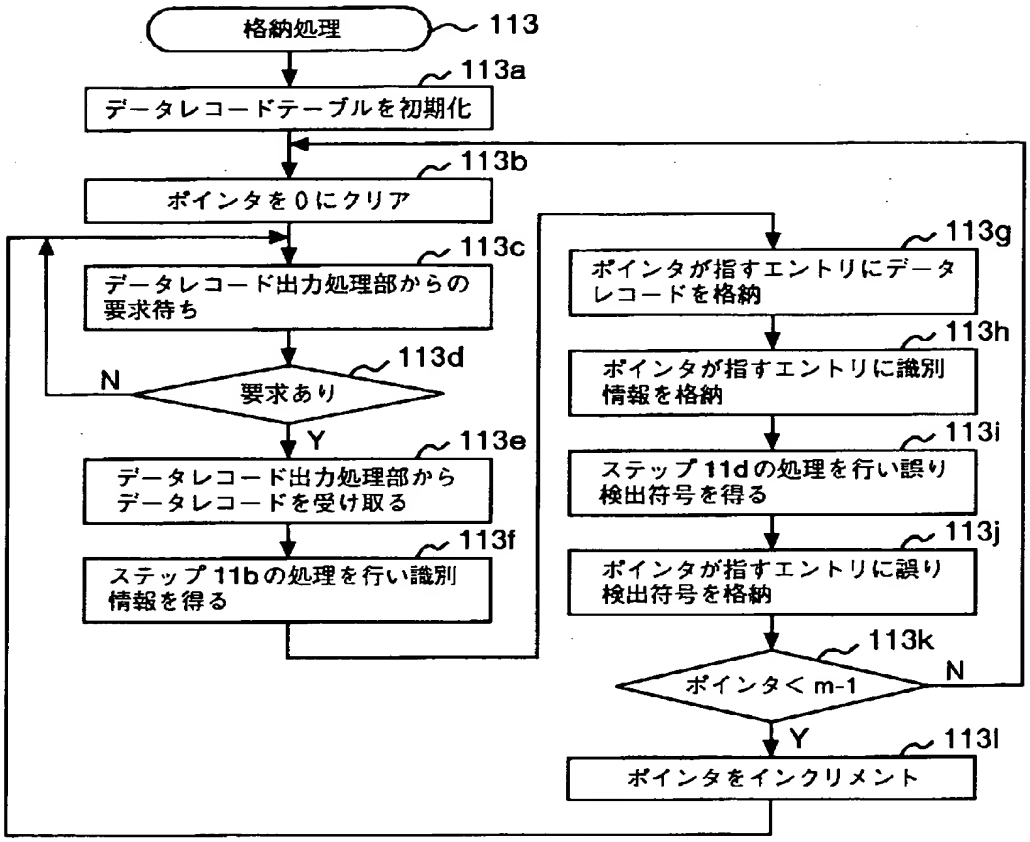
【図１４】

図　１４


第１のプログラム
110
データレコード出力処理部：
１個のデータレコードを出力し格納処理部に格納を依頼する
11a
格納処理部：
識別情報出力処理部を起動する
11b
識別情報出力処理部：
識別情報を返す
11c
格納処理部：
ステップ11aのデータレコードとステップ11cの識別情報とを組にして、誤り検出符号生成処理部に渡す
11d
誤り検出符号生成処理部：
渡された情報に対する誤り検出符号を返す
11e
格納処理部：
ステップ11aのデータレコードとステップ11cの識別情報、ステップ11eの誤り検出符号を組にして、ポインタの指すエントリに格納する
11f
格納処理部：
ポインタをインクリメント(ラップ)する
11h

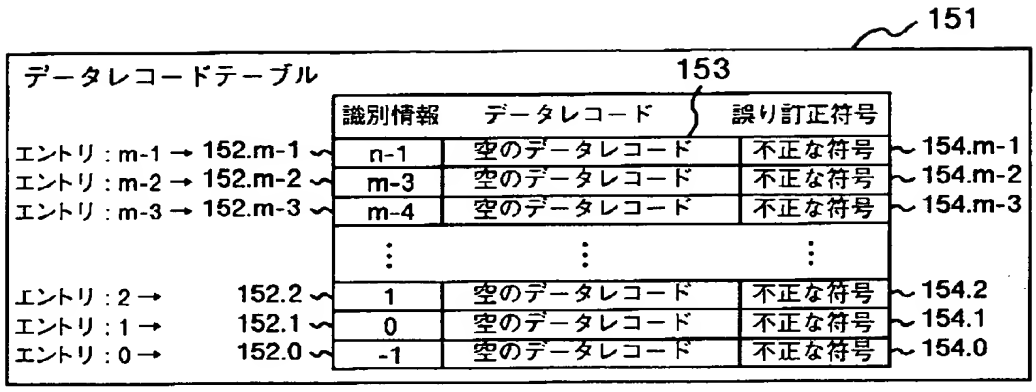
【図１５】

図　１５


格納処理
113
113a
データレコードテーブルを初期化
113b
ポインタを０にクリア
113c
データレコード出力処理部からの要求待ち
113d
要求あり
N
Y
113e
データレコード出力処理部からデータレコードを受け取る
113f
ステップ11bの処理を行い識別情報を得る
113g
ポインタが指すエントリにデータレコードを格納
113h
ポインタが指すエントリに識別情報を格納
113i
ステップ11dの処理を行い誤り検出符号を得る
113j
ポインタが指すエントリに誤り検出符号を格納
113k
ポインタ＜m-1
N
Y
113l
ポインタをインクリメント

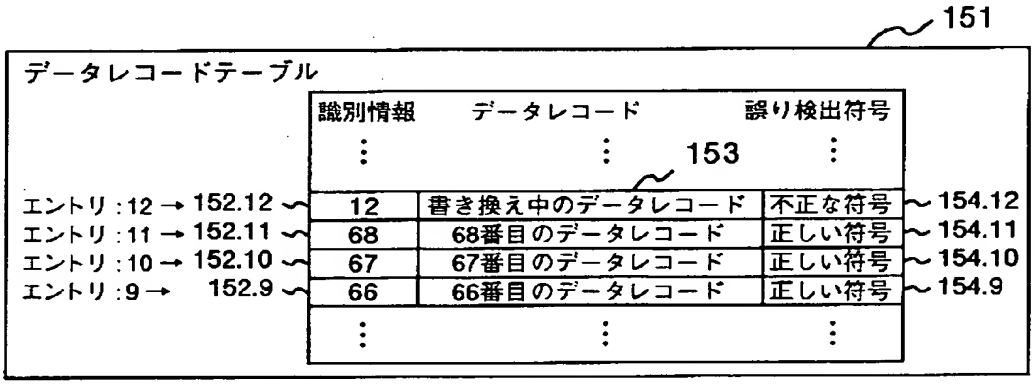
【図１６】

図　１６



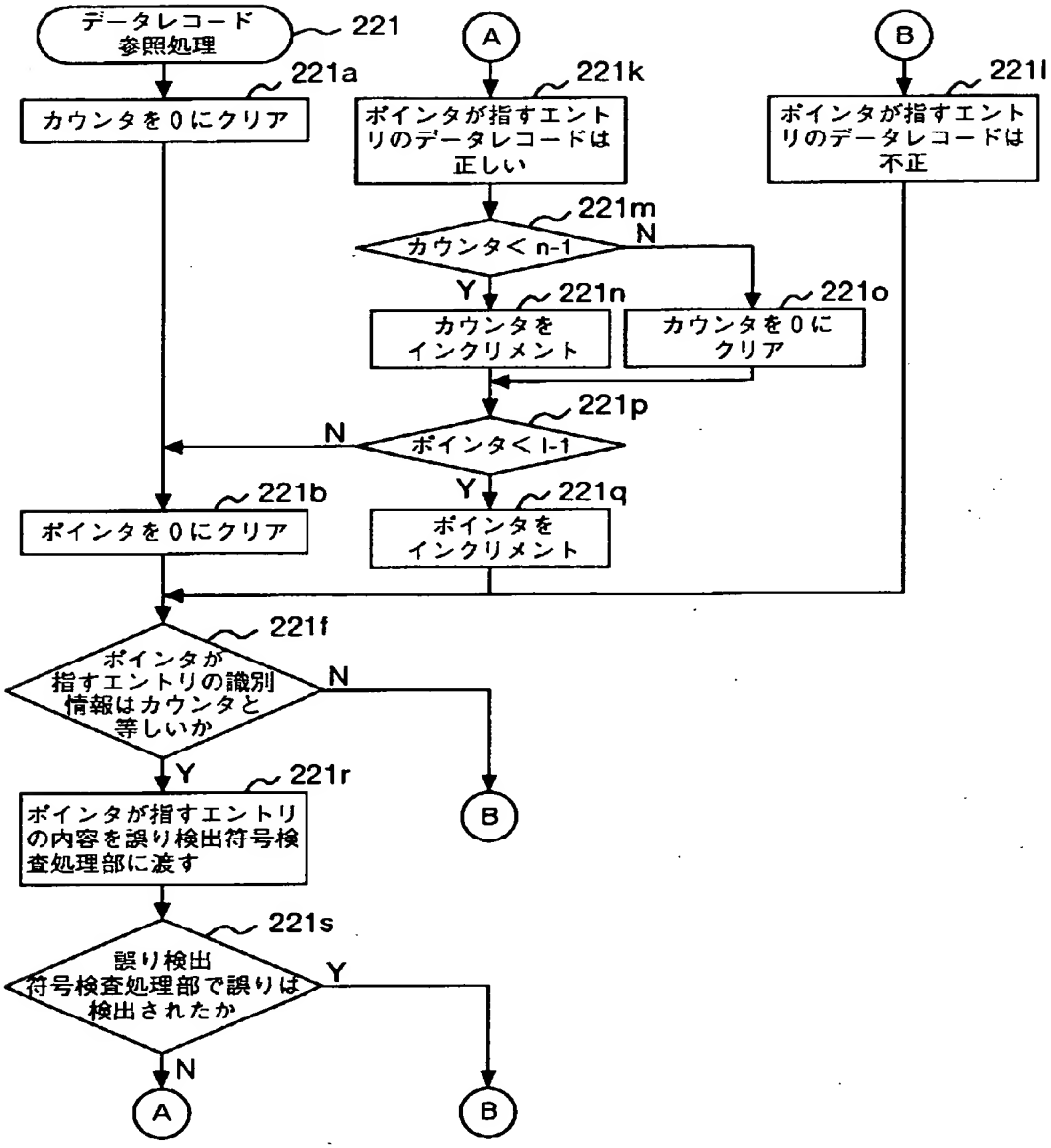
【図１７】

図　１７



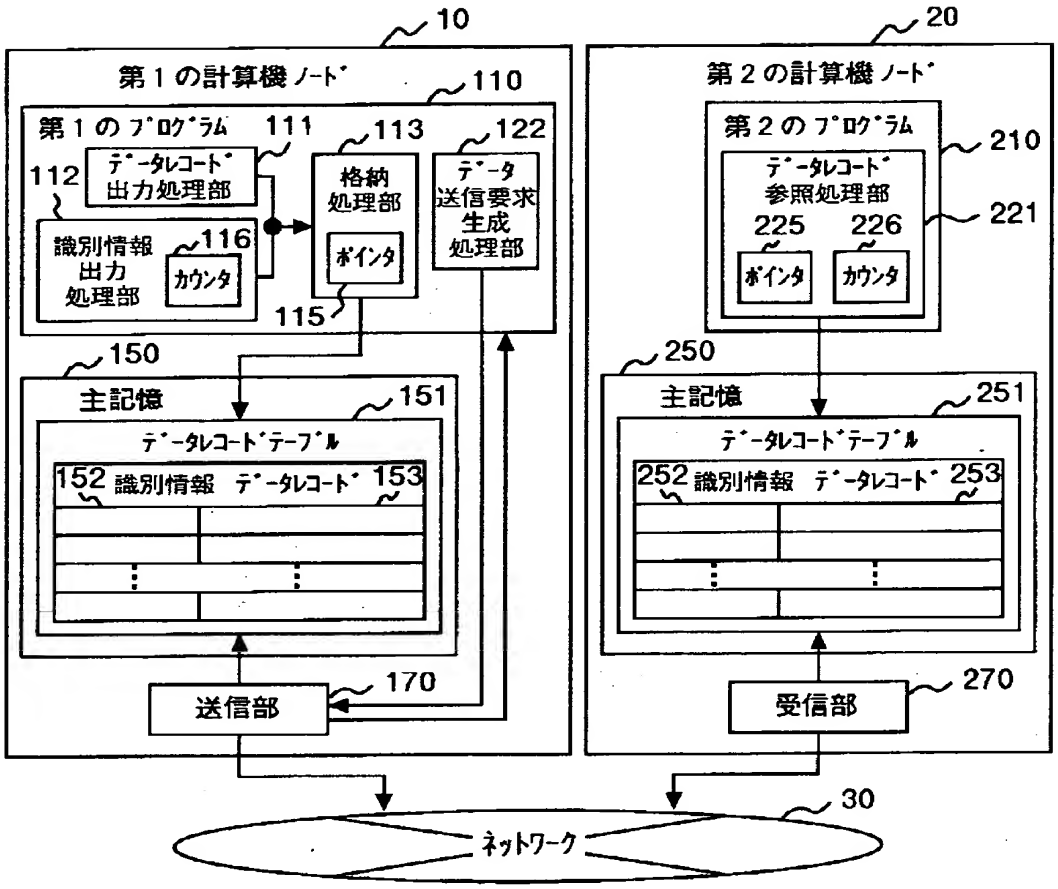
【図１８】

図　１８


データレコード参照処理
221
221a
カウンタを０にクリア
A
221k
ポインタが指すエントリのデータレコードは正しい
B
221l
ポインタが指すエントリのデータレコードは不正
221m
カウンタ＜ n-1
N
Y
221n
カウンタをインクリメント
221o
カウンタを０にクリア
221p
ポインタ＜ l-1
N
Y
221b
ポインタを０にクリア
221q
ポインタをインクリメント
221f
ポインタが指すエントリの識別情報はカウンタと等しいか
N
Y
221r
ポインタが指すエントリの内容を誤り検出符号検査処理部に渡す
B
221s
誤り検出符号検査処理部で誤りは検出されたか
Y
N
A
B

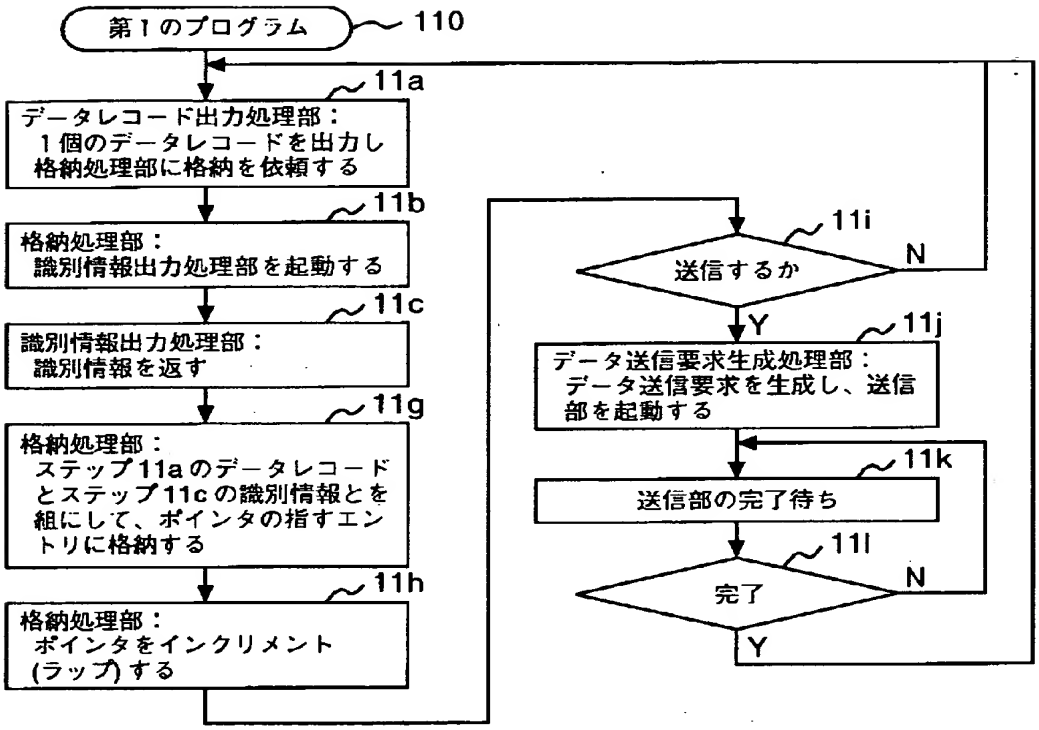
【図１９】

図　１９


10
第１の計算機ノード
110
第１のプログラム
111
112
データレコード出力処理部
113
格納処理部
ポインタ
122
データ送信要求生成処理部
識別情報出力処理部
116
カウンタ
115
150
主記憶
151
データレコードテーブル
152 識別情報
データレコード 153
送信部
170
20
第２の計算機ノード
第２のプログラム
210
データレコード参照処理部
221
225
226
ポインタ
カウンタ
250
主記憶
251
データレコードテーブル
252 識別情報
データレコード 253
受信部
270
30
ネットワーク

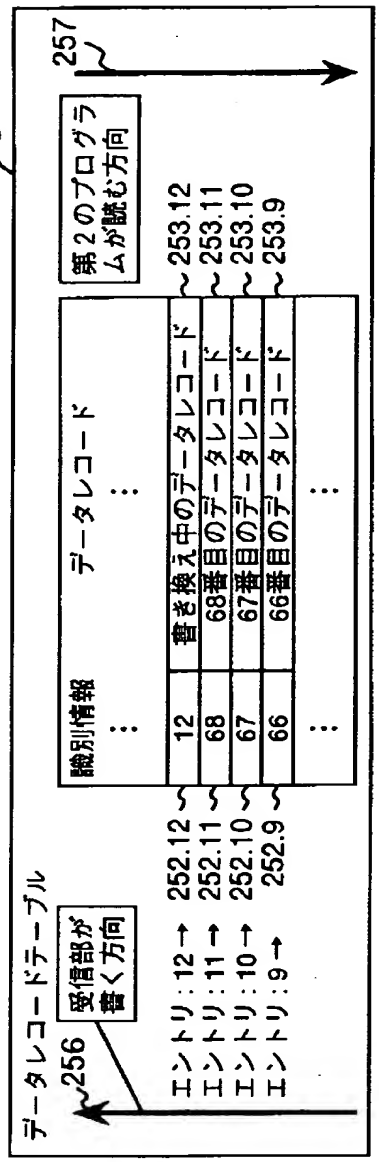
【図２０】

図　２０


第１のプログラム
110
11a
データレコード出力処理部：
１個のデータレコードを出力し
格納処理部に格納を依頼する
11b
格納処理部：
識別情報出力処理部を起動する
11c
識別情報出力処理部：
識別情報を返す
11g
格納処理部：
ステップ11aのデータレコード
とステップ11cの識別情報とを
組にして、ポインタの指すエン
トリに格納する
11h
格納処理部：
ポインタをインクリメント
(ラップ)する
11i
送信するか
N
Y
11j
データ送信要求生成処理部：
データ送信要求を生成し、送信
部を起動する
11k
送信部の完了待ち
11l
完了
N
Y

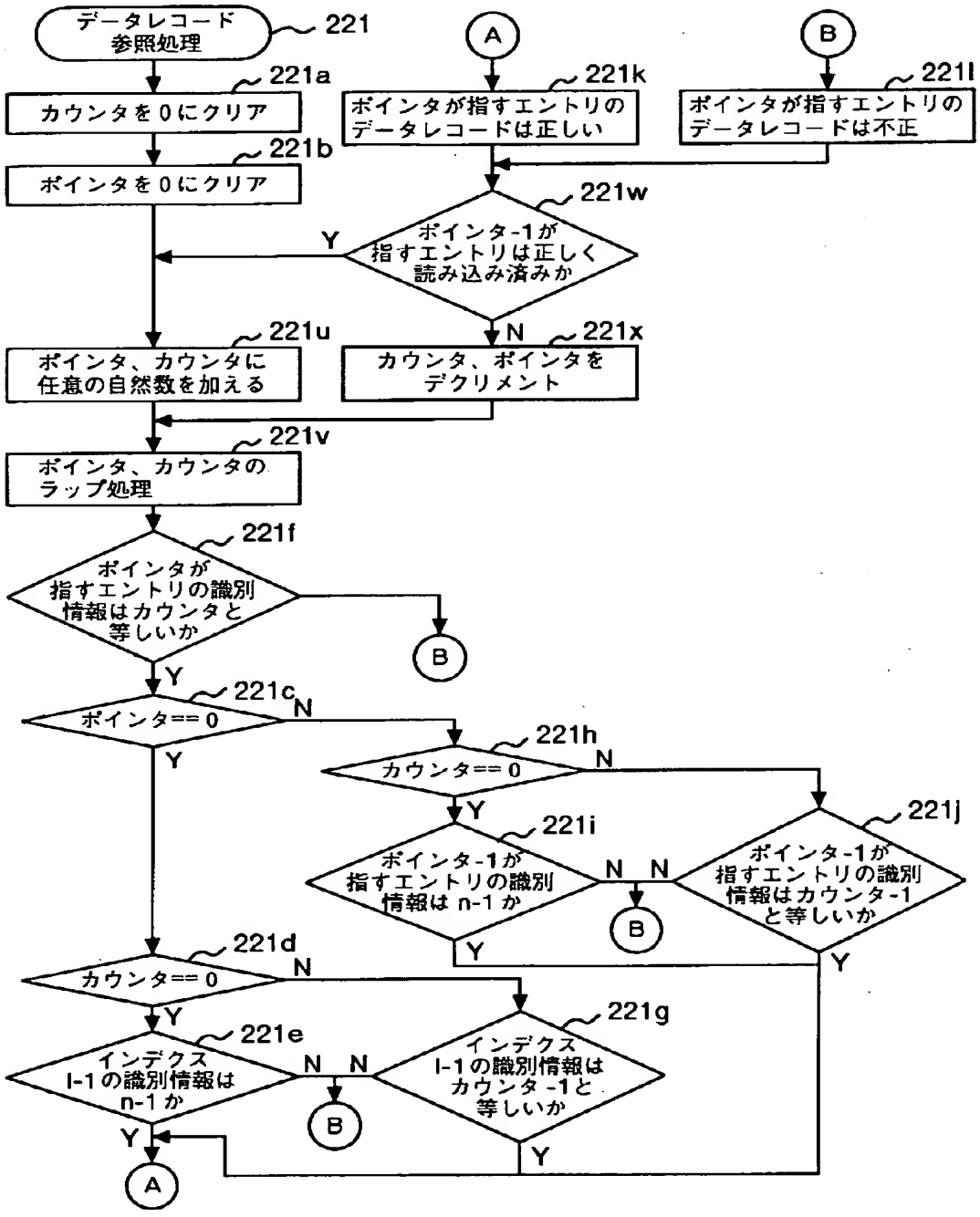
【図２１】

図　２１


251
257
データレコードテーブル
256
受信部が書く方向
エントリ：12→ 252.12
エントリ：11→ 252.11
エントリ：10→ 252.10
エントリ：9→ 252.9
識別情報
データレコード
第２のプログラムが読む方向
12
68
67
66
書き換え中のデータレコード
68番目のデータレコード
67番目のデータレコード
66番目のデータレコード
253.12
253.11
253.10
253.9

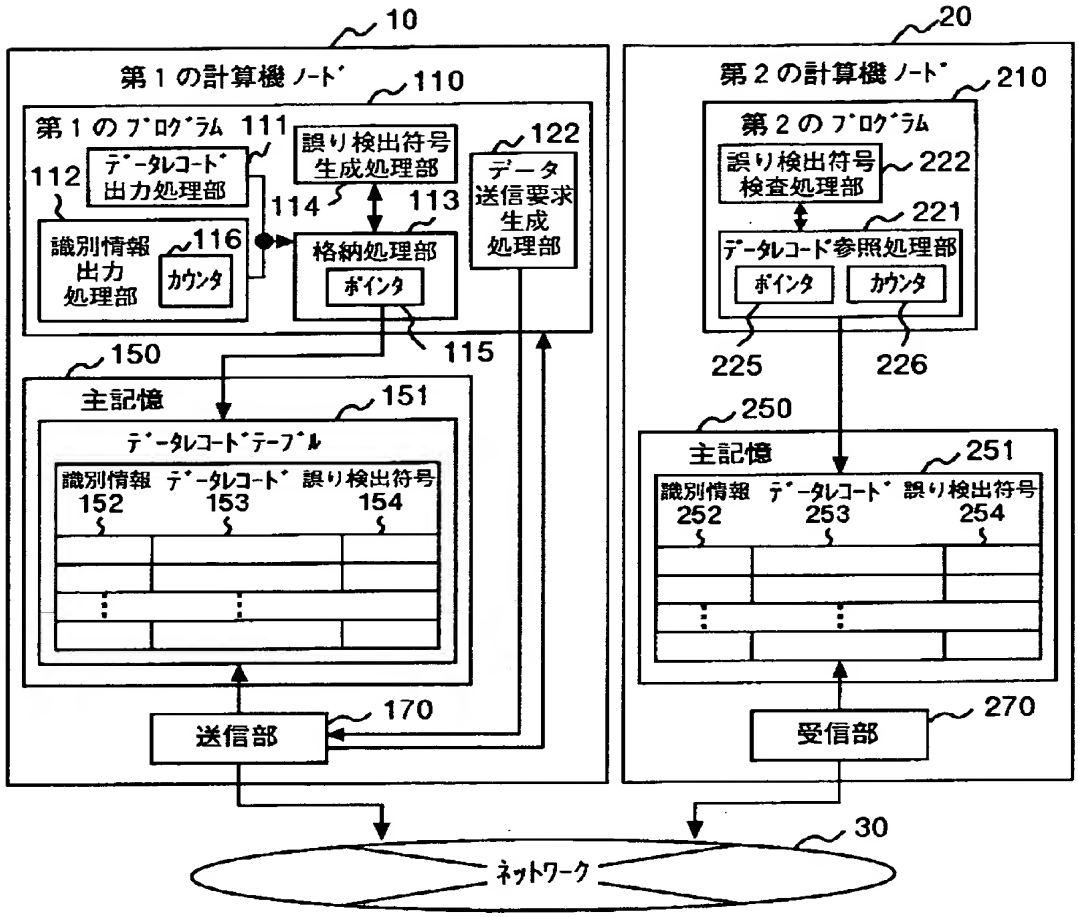
【図２２】

図　２２


データレコード参照処理
221
221a
カウンタを０にクリア
221b
ポインタを０にクリア
A
221k
ポインタが指すエントリのデータレコードは正しい
B
221l
ポインタが指すエントリのデータレコードは不正
221w
ポインタ-1が指すエントリは正しく読み込み済みか
Y
N
221u
ポインタ、カウンタに任意の自然数を加える
221x
カウンタ、ポインタをデクリメント
221v
ポインタ、カウンタのラップ処理
221f
ポインタが指すエントリの識別情報はカウンタと等しいか
B
Y
221c
ポインタ== 0
N
Y
221h
カウンタ== 0
N
Y
221i
ポインタ-1が指すエントリの識別情報は n-1 か
N N
B
Y
221j
ポインタ-1が指すエントリの識別情報はカウンタ-1と等しいか
Y
221d
カウンタ== 0
N
Y
221e
インデクス I-1の識別情報は n-1 か
N N
B
221g
インデクス I-1の識別情報はカウンタ -1と等しいか
Y
Y
A

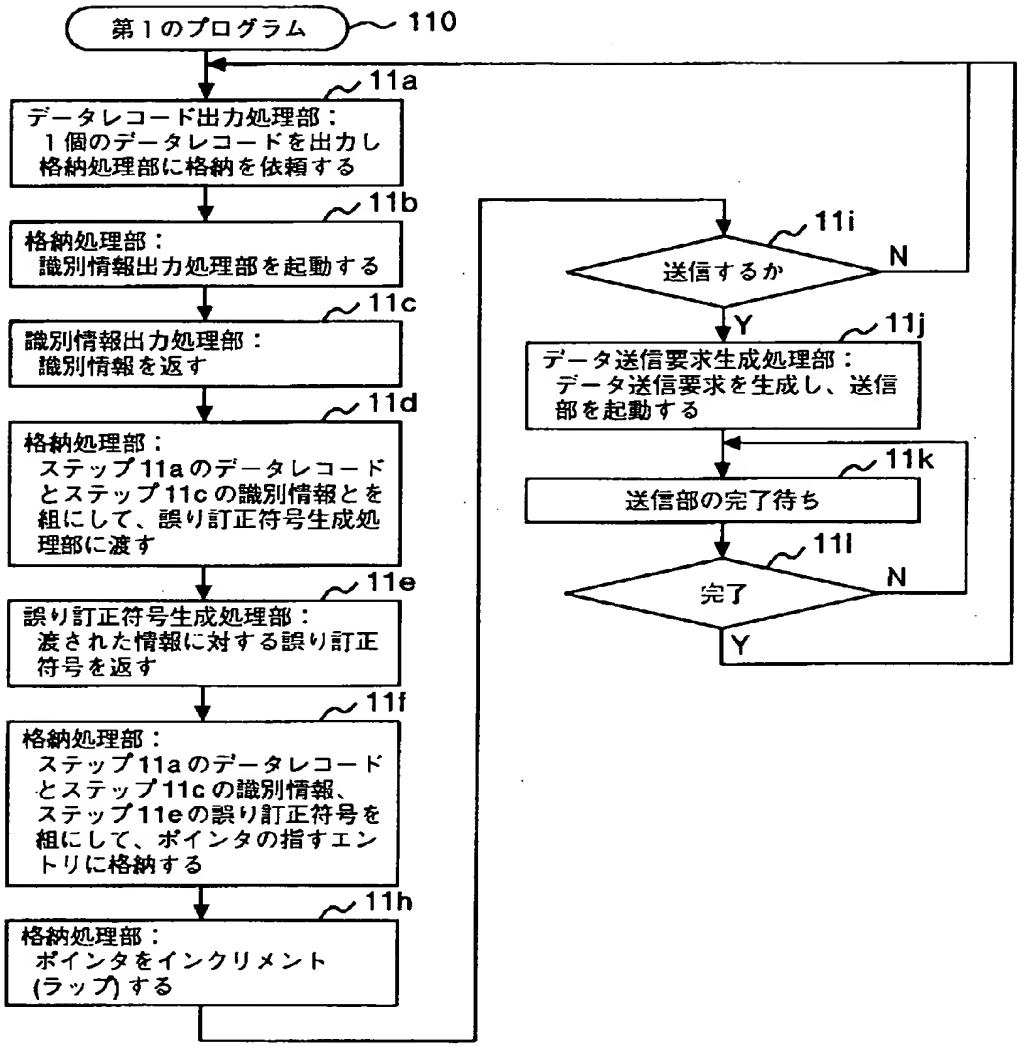
【図２３】

図　２３


10
第１の計算機ノード
110
第１のプログラム
111
データレコード出力処理部
112
誤り検出符号生成処理部
122
データ送信要求生成処理部
114
113
識別情報出力処理部
116
カウンタ
格納処理部
ポインタ
115
150
主記憶
151
データレコードテーブル
識別情報 152
データレコード 153
誤り検出符号 154
170
送信部
20
第２の計算機ノード
210
第２のプログラム
誤り検出符号検査処理部
222
221
データレコード参照処理部
ポインタ
カウンタ
225
226
250
主記憶
251
識別情報 252
データレコード 253
誤り検出符号 254
270
受信部
30
ネットワーク

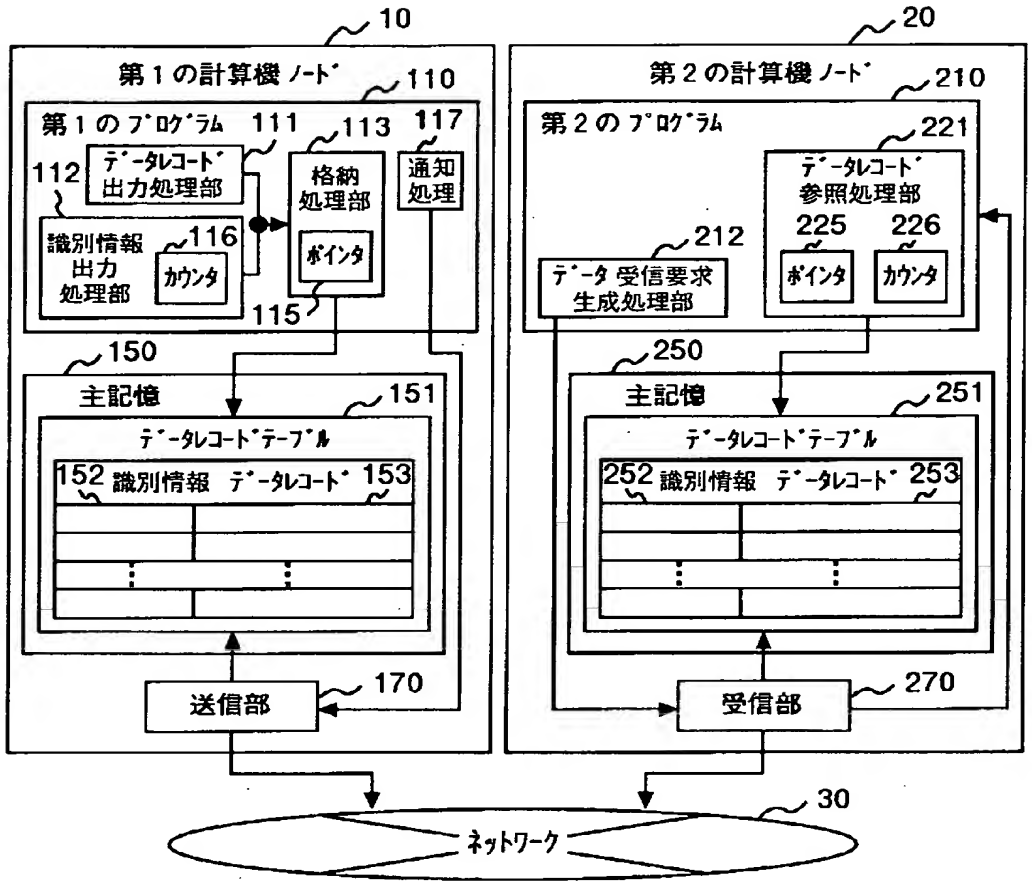
【図２４】

図　２４


第１のプログラム
110
11a
データレコード出力処理部：
１個のデータレコードを出力し
格納処理部に格納を依頼する
11b
格納処理部：
識別情報出力処理部を起動する
11c
識別情報出力処理部：
識別情報を返す
11d
格納処理部：
ステップ11aのデータレコード
とステップ11cの識別情報とを
組にして、誤り訂正符号生成処
理部に渡す
11e
誤り訂正符号生成処理部：
渡された情報に対する誤り訂正
符号を返す
11f
格納処理部：
ステップ11aのデータレコード
とステップ11cの識別情報、
ステップ11eの誤り訂正符号を
組にして、ポインタの指すエン
トリに格納する
11h
格納処理部：
ポインタをインクリメント
(ラップ)する
11i
送信するか
N
Y
11j
データ送信要求生成処理部：
データ送信要求を生成し、送信
部を起動する
11k
送信部の完了待ち
11l
完了
N
Y

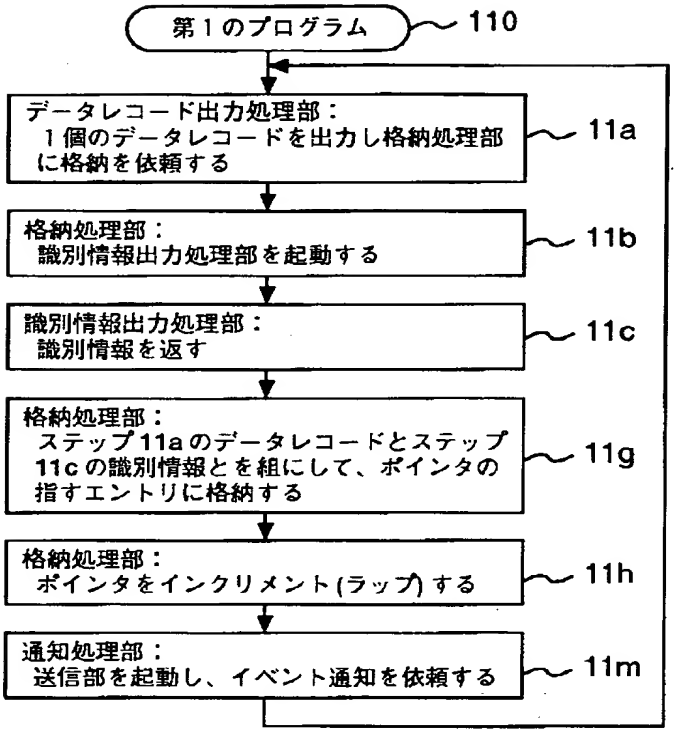
【図２５】

図　２５


10
第１の計算機ノード
110
第１のプログラム
111
113
117
112
データレコード出力処理部
格納処理部
通知処理
識別情報出力処理部
116
カウンタ
ポインタ
115
150
主記憶
151
データレコードテーブル
152
識別情報
データレコード
153
送信部
170
20
第２の計算機ノード
210
第２のプログラム
221
データレコード参照処理部
225
226
212
データ受信要求生成処理部
ポインタ
カウンタ
250
主記憶
251
データレコードテーブル
252
識別情報
データレコード
253
受信部
270
30
ネットワーク

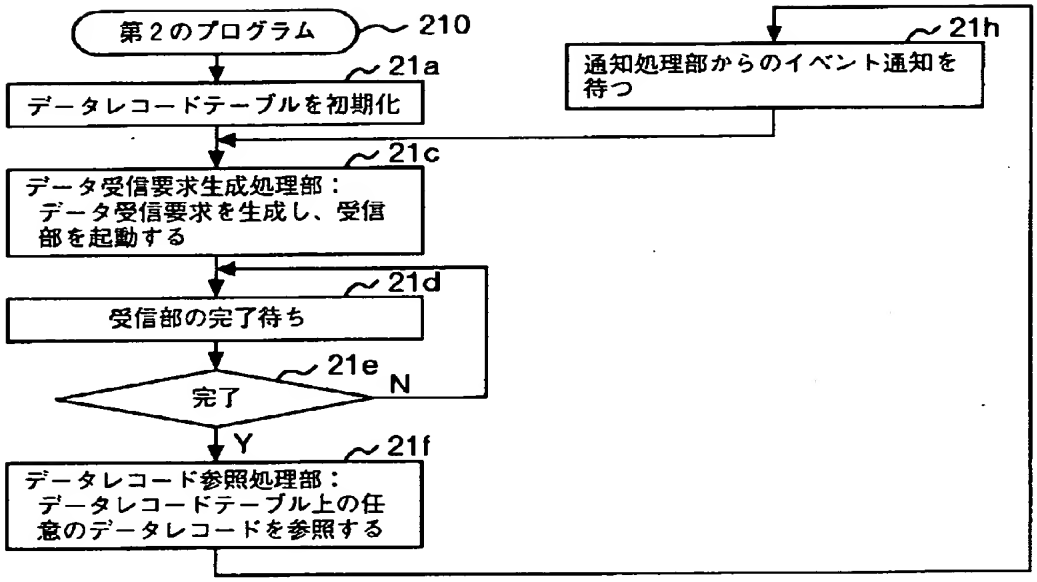
【図２６】

図　２６


第１のプログラム
110
データレコード出力処理部：
１個のデータレコードを出力し格納処理部に格納を依頼する
11a
格納処理部：
識別情報出力処理部を起動する
11b
識別情報出力処理部：
識別情報を返す
11c
格納処理部：
ステップ11aのデータレコードとステップ11cの識別情報とを組にして、ポインタの指すエントリに格納する
11g
格納処理部：
ポインタをインクリメント(ラップ)する
11h
通知処理部：
送信部を起動し、イベント通知を依頼する
11m

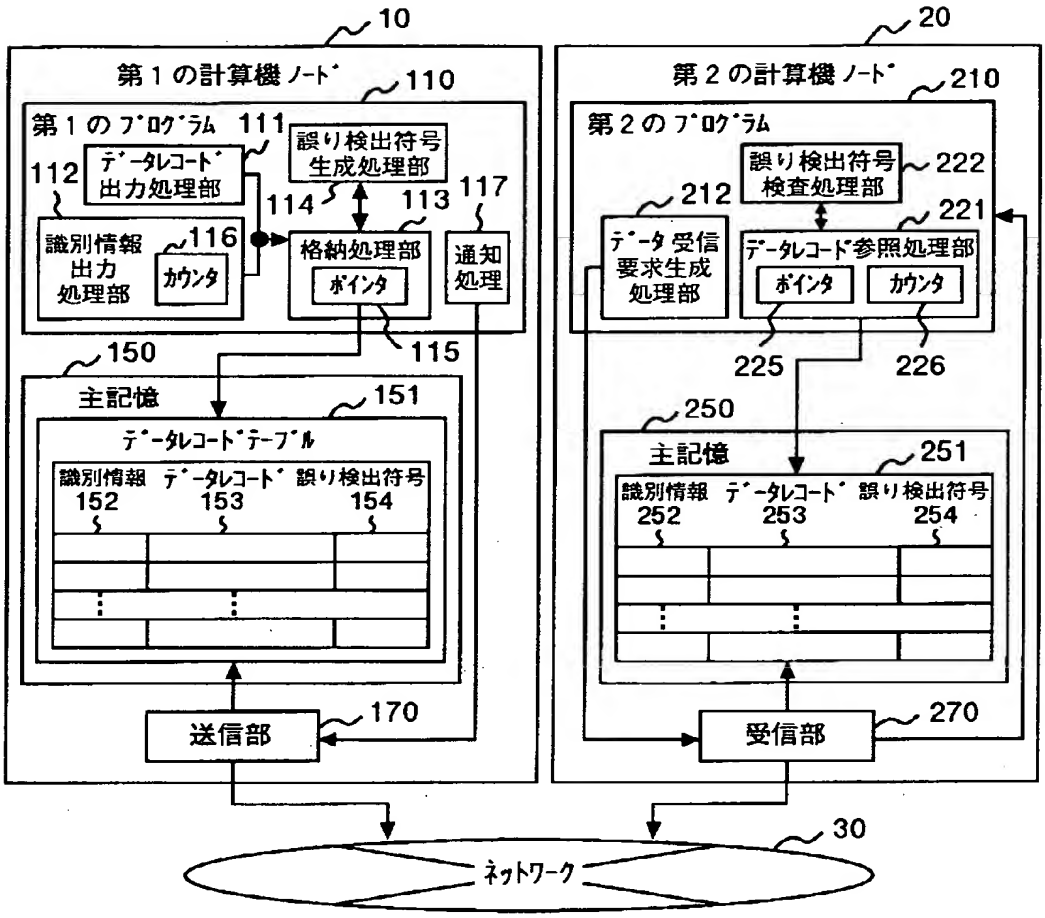
【図２７】

図　２７


第２のプログラム
210
21a
データレコードテーブルを初期化
21c
データ受信要求生成処理部：
データ受信要求を生成し、受信部を起動する
21d
受信部の完了待ち
21e
完了
N
Y
21f
データレコード参照処理部：
データレコードテーブル上の任意のデータレコードを参照する
21h
通知処理部からのイベント通知を待つ

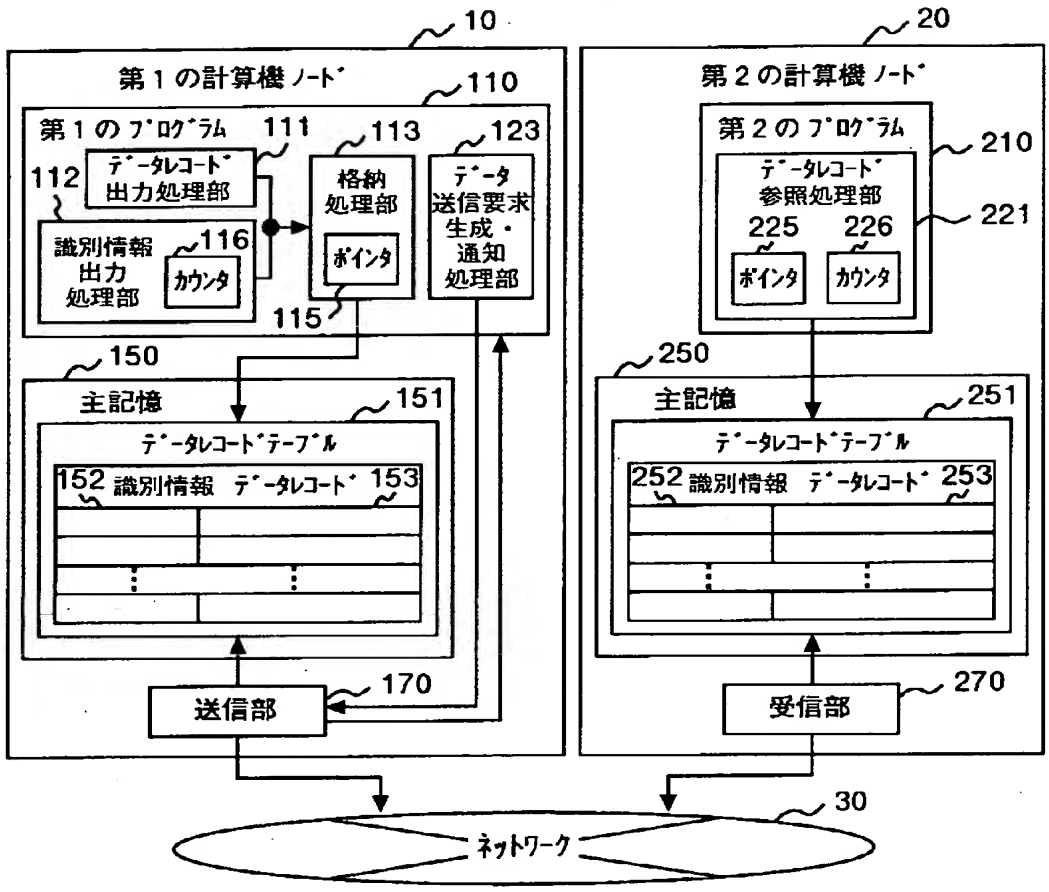
【図２８】

図　２８


10
第１の計算機ノード
110
第１のプログラム
111
誤り検出符号生成処理部
112
データレコード出力処理部
114
113
117
識別情報出力処理部
116
カウンタ
格納処理部
ポインタ
通知処理
115
150
主記憶
151
データレコードテーブル
識別情報 152
データレコード 153
誤り検出符号 154
送信部
170
20
第２の計算機ノード
210
第２のプログラム
誤り検出符号検査処理部
222
212
221
データ受信要求生成処理部
データレコード参照処理部
ポインタ
カウンタ
225
226
250
主記憶
251
識別情報 252
データレコード 253
誤り検出符号 254
受信部
270
30
ネットワーク

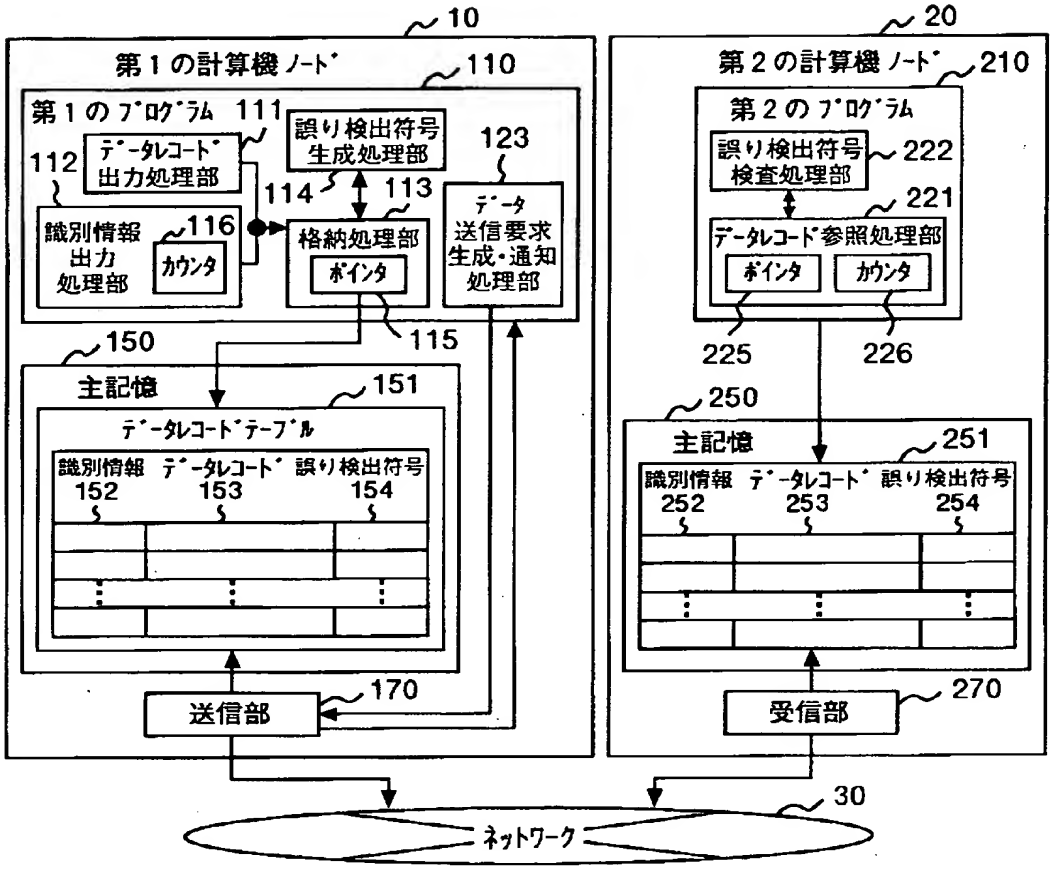
【図２９】

図　２９


10
第１の計算機ノード
110
第１のプログラム
111
113
123
112
データレコード出力処理部
格納処理部
データ送信要求生成・通知処理部
識別情報出力処理部
116
カウンタ
ポインタ
115
150
主記憶
151
データレコードテーブル
152 識別情報 データレコード 153
170
送信部
20
第２の計算機ノード
第２のプログラム
210
データレコード参照処理部
221
225
226
ポインタ
カウンタ
250
主記憶
251
データレコードテーブル
252 識別情報 データレコード 253
270
受信部
30
ネットワーク

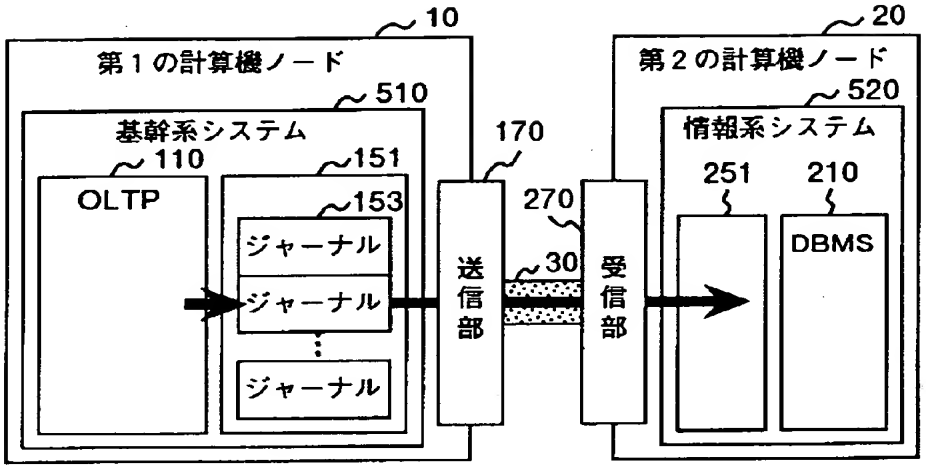
【図３０】

図　３０


10
第１の計算機ﾉｰﾄﾞ
110
第１のﾌﾟﾛｸﾞﾗﾑ
111
ﾃﾞｰﾀﾚｺｰﾄﾞ出力処理部
112
誤り検出符号生成処理部
123
識別情報出力処理部
116
ｶｳﾝﾀ
114
113
格納処理部
ﾎﾟｲﾝﾀ
ﾃﾞｰﾀ送信要求生成・通知処理部
115
150
主記憶
151
ﾃﾞｰﾀﾚｺｰﾄﾞﾃｰﾌﾞﾙ
識別情報 152
ﾃﾞｰﾀﾚｺｰﾄﾞ 153
誤り検出符号 154
送信部
170
20
第２の計算機ﾉｰﾄﾞ
210
第２のﾌﾟﾛｸﾞﾗﾑ
誤り検出符号検査処理部
222
221
ﾃﾞｰﾀﾚｺｰﾄﾞ参照処理部
ﾎﾟｲﾝﾀ
ｶｳﾝﾀ
225
226
250
主記憶
251
識別情報 252
ﾃﾞｰﾀﾚｺｰﾄﾞ 253
誤り検出符号 254
受信部
270
30
ネットワーク

【図３１】

図　３１


10
第１の計算機ノード
510
基幹系システム
110
OLTP
151
153
ジャーナル
ジャーナル
ジャーナル
170
送信部
270
30
受信部
20
第２の計算機ノード
520
情報系システム
251
210
DBMS

【書類名】　要約書

【要約】

【課題】従来の２つの計算機ノード間のデータ受け渡し方法では、データ転送を行うプログラム間の待ち合わせのオーバヘッドが大きい。

【解決手段】データを送信する第１のプログラムが、任意の時間間隔でデータを主記憶の領域に格納する。データを受信する第２のプログラムは、任意の時間間隔で上記領域をＲＤＭＡを用いて、参照する。または、データを受信する第２のプログラムは、任意の時間間隔で主記憶の領域を参照する。データを送信する第１のプログラムは、任意の時間間隔でＲＤＭＡを用いてデータを上記領域に格納する。さらに、書き込みと読み出しのすれ違いを検出するため、各レコードに識別子を付け、識別子とレコード本体のアクセス順序を書き込みと読み出しで逆にする。あるいは各レコードに識別子と誤り検出符号を付加する。

【選択図】図１

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所